Προγραμματισμός & Συστήματα στον Παγκόσμιο Ιστό  
*Εργαστηριακή Άσκηση Ακαδημαϊκού Έτους 2024-25*

Θέμα: Σύστημα Υποστήριξης Διπλωματικών Εργασιών

Ονοματεπώνυμο:  
Δερβένης Ιάσονας  
Κολόι Νίκος  
Ξένου Εμμανουέλα  
Τούντας Δημήτρης   
Αριθμός Μητρώου:   
1097437  
1063429  
1054286  
1093504Έτος Εισαγωγής:  
2021-22  
2017-18  
2016-17  
2021-22

ΠΕΡΙΕΧΟΜΕΝΑ

# 1. Εισαγωγή

Στόχος της εργασίας μας είναι η: «*Ανάπτυξη ενός συνεργατικού συστήματος για την υποστήριξη μιας κοινής διοικητικής διαδικασίας ενός πανεπιστημίου, και συγκεκριμένα η εκπόνηση της διπλωματικής εργασίας»*. Η περιγραφή της λειτουργίας της πλατφόρμας διαχείρισης διπλωματικών μεταξύ φοιτητών και καθηγητών, γίνεται μέσα από screenshots και συνοπτικές επεξηγήσεις (όπου αυτές είναι αναγκαίες)

# 2. Περιγραφή Υλοποίησης

**Backend**

Αναπτύχθηκε σε Node.js με χρήση του Express framework. Χρησιμοποιήθηκε η βιβλιοθήκη mysql2 για τη σύνδεση με MySQL και την εκτέλεση queries. Ενσωματώθηκαν middleware όπως cors, express.json και express.urlencoded. Η λογική οργανώθηκε σε διαφορετικά routes (login, student, professor, admin, topics, announcements).

**Frontend**

Βασίστηκε σε HTML5 και CSS, με χρήση του Bootstrap framework για responsive σχεδιασμό. Η JavaScript υλοποιεί τη διαδραστικότητα με AJAX, modals, counters και δυναμικά στοιχεία. Για την απεικόνιση δεδομένων χρησιμοποιήθηκε η βιβλιοθήκη Chart.js. Η οργάνωση έγινε σε views, protected\_views και public.

**Βάση Δεδομένων - Αρχικοποίηση Συστήματος**

Η αποθήκευση των δεδομένων έγινε σε MySQL (xampp). Στον φάκελο mysql\_server υπάρχουν τα SQL scripts για τη δημιουργία και αρχικοποίηση της βάσης. Υλοποιήθηκαν οι πίνακες Users, Diplomas, Committees, Invitations, Presentations, Grades, Files, σύμφωνα με τις απαιτήσεις της εκφώνησης

Για την αρχικοποίηση του συστήματος υλοποιήσαμε μηχανισμό φόρτωσης αρχικών δεδομένων μέσω του αρχείου **data.json**.

* Το αρχείο **config.js** περιέχει τη συνάρτηση **insertData()**, η οποία εισάγει τα δεδομένα στη βάση.
* Η συνάρτηση εισάγεται ως module στον **server.js** και εκτελείται με την εκκίνηση του server.
* Πριν από κάθε εισαγωγή γίνεται έλεγχος για **διπλοτυπίες**, ώστε να μην προστεθούν ξανά τα ίδια δεδομένα.
* Τα δεδομένα του **data.json** προήλθαν από το **κοινό αποθετήριο** που δίνεται στην εκφώνηση. Στη συνέχεια έγιναν οι απαραίτητες **επεξεργασίες, προσθήκες και τροποποιήσεις**, ώστε να καλύπτονται όλες οι πιθανές καταστάσεις διπλωματικών και να αναδεικνύονται οι λειτουργίες του συστήματος (π.χ. login με διαφορετικούς ρόλους, αναθέσεις, αλλαγές κατάστασης).

# 3. Entity Relationship Diagram

Η βάση δεδομένων του συστήματος είναι σχεσιακή και σχεδιάστηκε με γνώμονα τη σαφή αποτύπωση όλων των λειτουργιών που απαιτούνται για τη διαχείριση διπλωματικών εργασιών.

**Κύριες Οντότητες**

**Καθηγητές, Φοιτητές, Γραμματείς**: τρεις ξεχωριστοί πίνακες χρηστών με primary key το id. Το email και το password χρησιμοποιούνται για login. Ο διαχωρισμός των ρόλων σε διαφορετικούς πίνακες επιτρέπει να διατηρούνται διαφορετικά χαρακτηριστικά ανά κατηγορία χρήστη.

**Διπλωματικές**: κεντρική οντότητα του συστήματος. Συνδέεται με έναν φοιτητή και έναν επιβλέποντα καθηγητή μέσω foreign keys. Περιλαμβάνει ιδιότητες όπως status (με enum τιμές), pdf\_path, start\_date, nimertis\_link, grading\_enabled και final\_grade.

**Προσκλήσεις**: συνδέονται με συγκεκριμένη διπλωματική και καθηγητή. Περιγράφουν την κατάσταση αποδοχής/απόρριψης μέσω του πεδίου status.

**Τριμελείς Επιτροπές**: συνδέονται με μία διπλωματική και περιέχουν τα ids των δύο μελών (ο τρίτος είναι ο επιβλέπων).

**Καταχωρήσεις Βαθμών**: κάθε διπλωματική συνδέεται με τρεις βαθμολογικές καταχωρήσεις, μία από κάθε μέλος της τριμελούς. Για κάθε καταχώρηση αποθηκεύονται τέσσερα κριτήρια αξιολόγησης και η κατάσταση finalized.

**Επισυναπτόμενα αρχεία/σύνδεσμοι**: κάθε διπλωματική μπορεί να έχει συνδεδεμένα αρχεία ή υπερσυνδέσμους. Καταγράφεται ο τύπος (type) και το αντίστοιχο path.

**Εξετάσεις**: αντιστοιχούν 1:1 με τις διπλωματικές. Καταγράφεται ο τρόπος εξέτασης (δια ζώσης ή εξ αποστάσεως), η τοποθεσία ή ο σύνδεσμος, και η ημερομηνία/ώρα.

**Αρχεία καταγραφών**: κάθε αλλαγή κατάστασης μιας διπλωματικής συνοδεύεται από καταχώρηση με λεπτομέρειες (π.χ. αριθμός πρωτοκόλλου, απόφαση ΓΣΤ, λόγος ακύρωσης).

**Ανακοινώσεις**: σχετίζονται 1:1 με μια διπλωματική και έχουν ημερομηνία δημοσίευσης.

**Σημειώσεις**: αφορούν σχόλια των καθηγητών της τριμελούς, συνδέονται με συγκεκριμένη διπλωματική και καθηγητή.

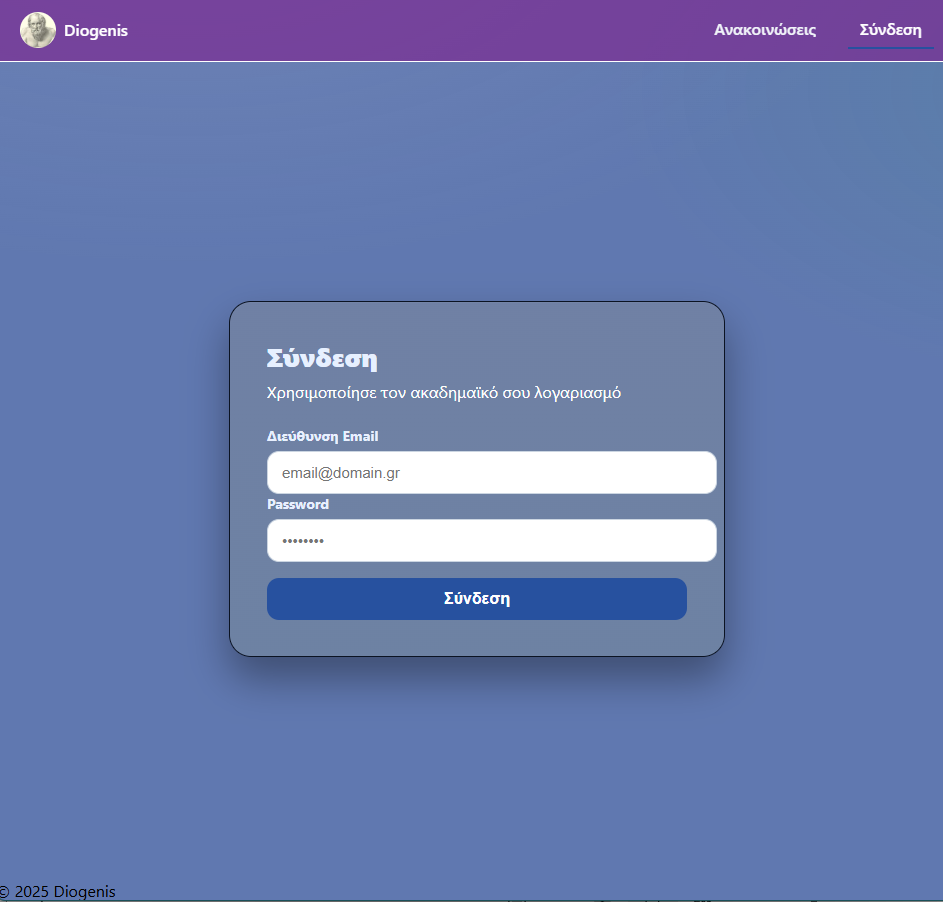
Επιλέξαμε να **μην** συγκεντρώσουμε όλους τους χρήστες σε έναν ενιαίο πίνακα users, ώστε:

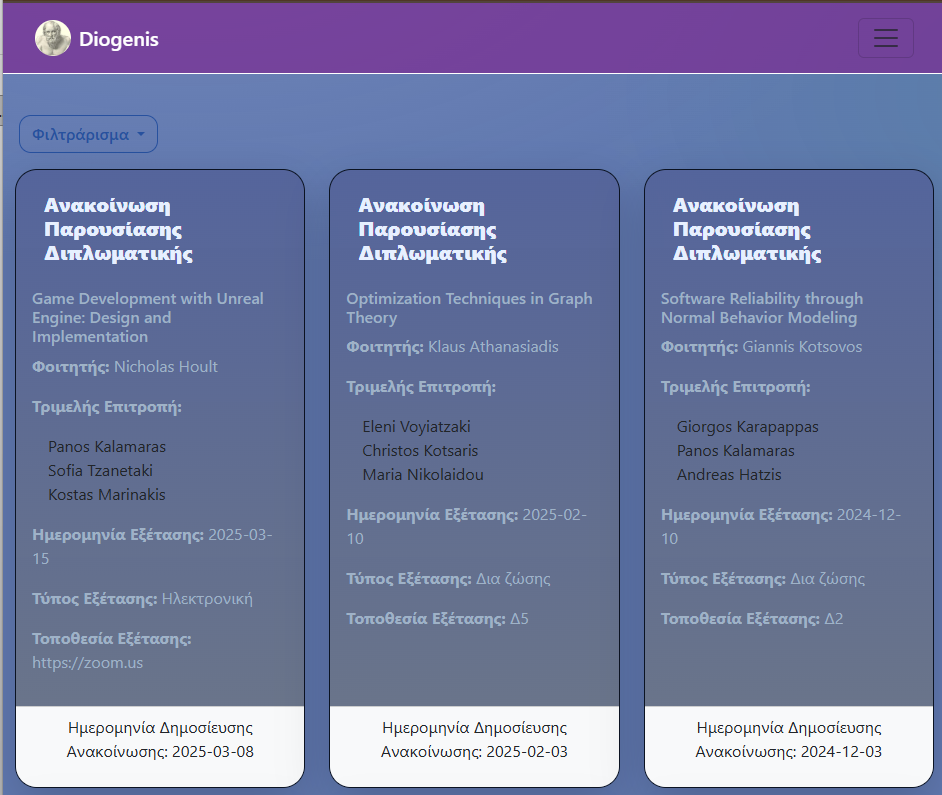
* να μπορούμε να αποθηκεύουμε διαφορετικά χαρακτηριστικά ανάλογα με τον τύπο χρήστη (φοιτητής, καθηγητής, γραμματεία) και
* να διευκολύνεται ο σαφής διαχωρισμός τους από το ίδιο το σύστημα.

**4. Παραδείγματα Χρήσης**

**Κοινά για όλους τους ρόλους**

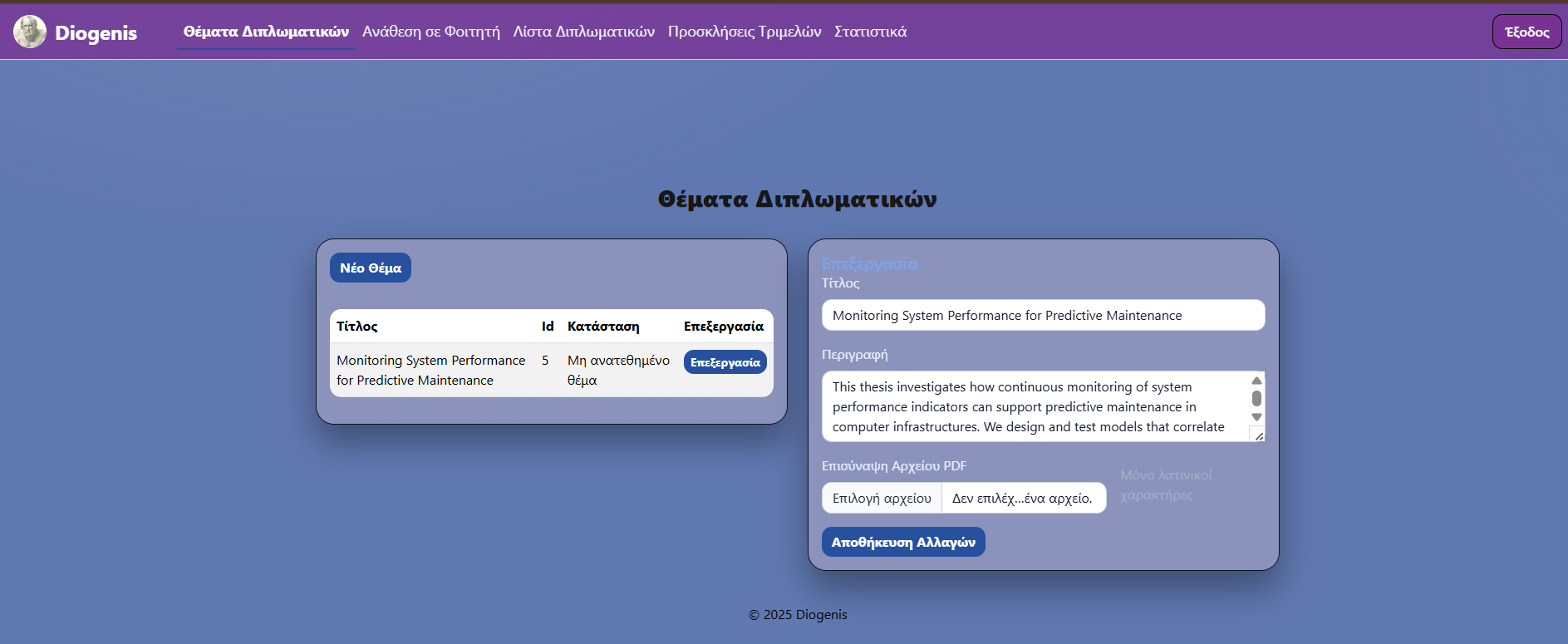
* **Login**

Ο χρήστης εισάγει username/password και αν είναι σωστά συνδέεται.  
Αν τα στοιχεία είναι λανθασμένα, απορρίπτεται η σύνδεση.

* ******Ανακοινώσεις**

**Διδάσκων**

* **Προβολή, Δημιουργία και Επεξεργασία Θεμάτων**

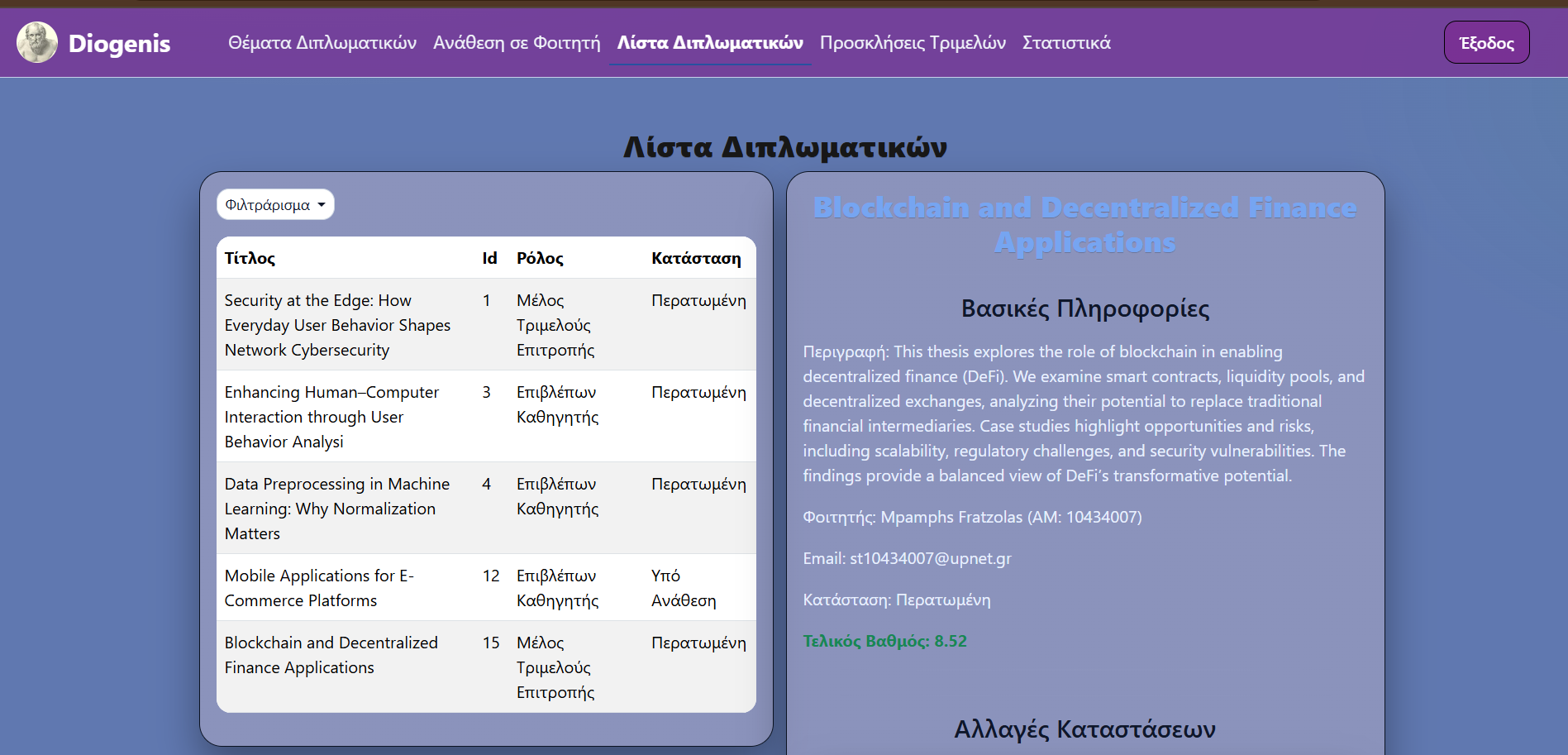
Βλέπει λίστα θεμάτων που έχει δημιουργήσει και μπορεί να επεξεργαστεί το θέμα. Επίσης μπορεί να καταχωρήσει νέο θέμα με τίτλο, περιγραφή, PDF.

* **Ανάθεση Θέματος σε Φοιτητή**

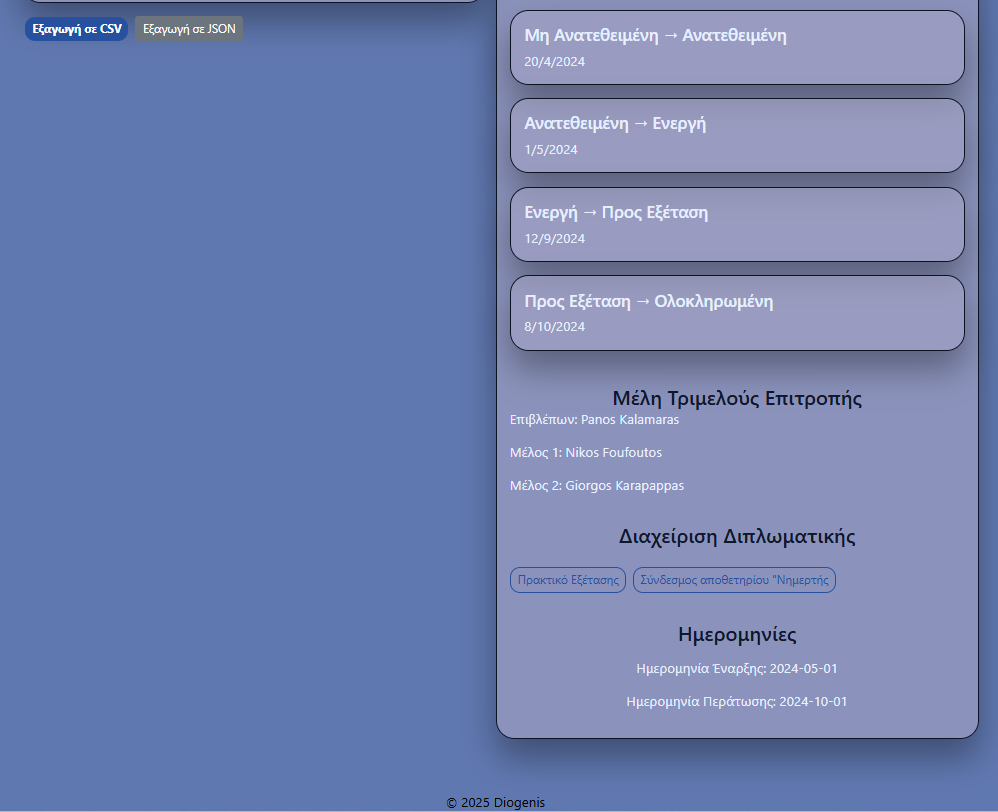
Εκχωρεί θέμα σε φοιτητή με βάση ΑΜ/όνομα. Κατοχύρωση προσωρινή μέχρι έγκριση τριμελούς. Μπορεί να αναιρέσει την ανάθεση πριν οριστικοποιηθεί.

* **Προβολή Λίστας Διπλωματικών**

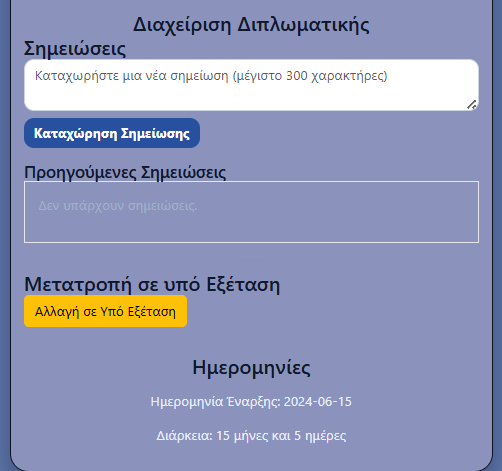
Βλέπει όλες τις ΔΕ στις οποίες έχει συμμετάσχει (ως επιβλέπων ή μέλος). Φιλτράρει με βάση κατάσταση και ρόλο. Για την επιλεγμένη Δ.Ε. βλέπει στοιχεία φοιτητή, θέμα, τριμελή, ιστορικό ενεργειών, βαθμό, συνδέσμους αρχείων Και μπορεί να εξάγει τα στοιχεία διπλωματικών σε CSV ή JSON

* Διαχείριση Δ.Ε. (ανάλογα με κατάσταση):

**Περατωμένη:**

****

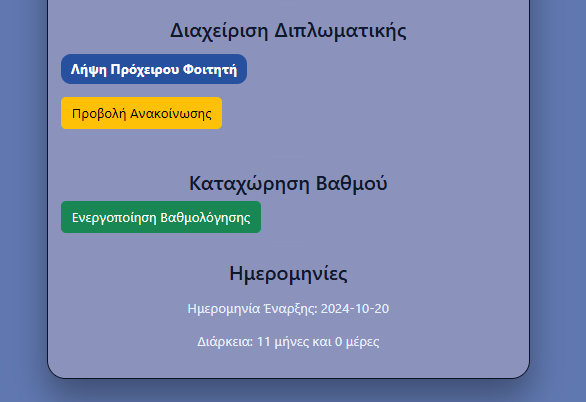
**Ενεργή:** Καταχωρεί σημειώσεις αν είναι επιβλέπων μπορεί να ακυρώσει ανάθεση ή να την αλλάξει σε «Υπό Εξέταση».



**Υπό ανάθεση**: Βλέπει μέλη και απαντήσεις· μπορεί να ακυρώσει ανάθεση.

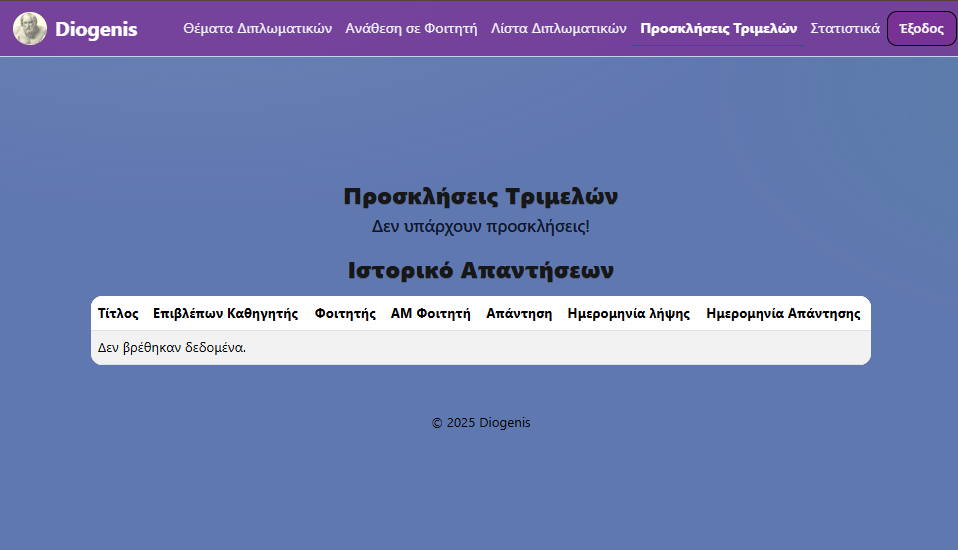


**Υπό εξέταση**: Βλέπει πρόχειρο κείμενο· ως επιβλέπων παράγει ανακοίνωση παρουσίασης· καταχωρεί/βλέπει βαθμούς.

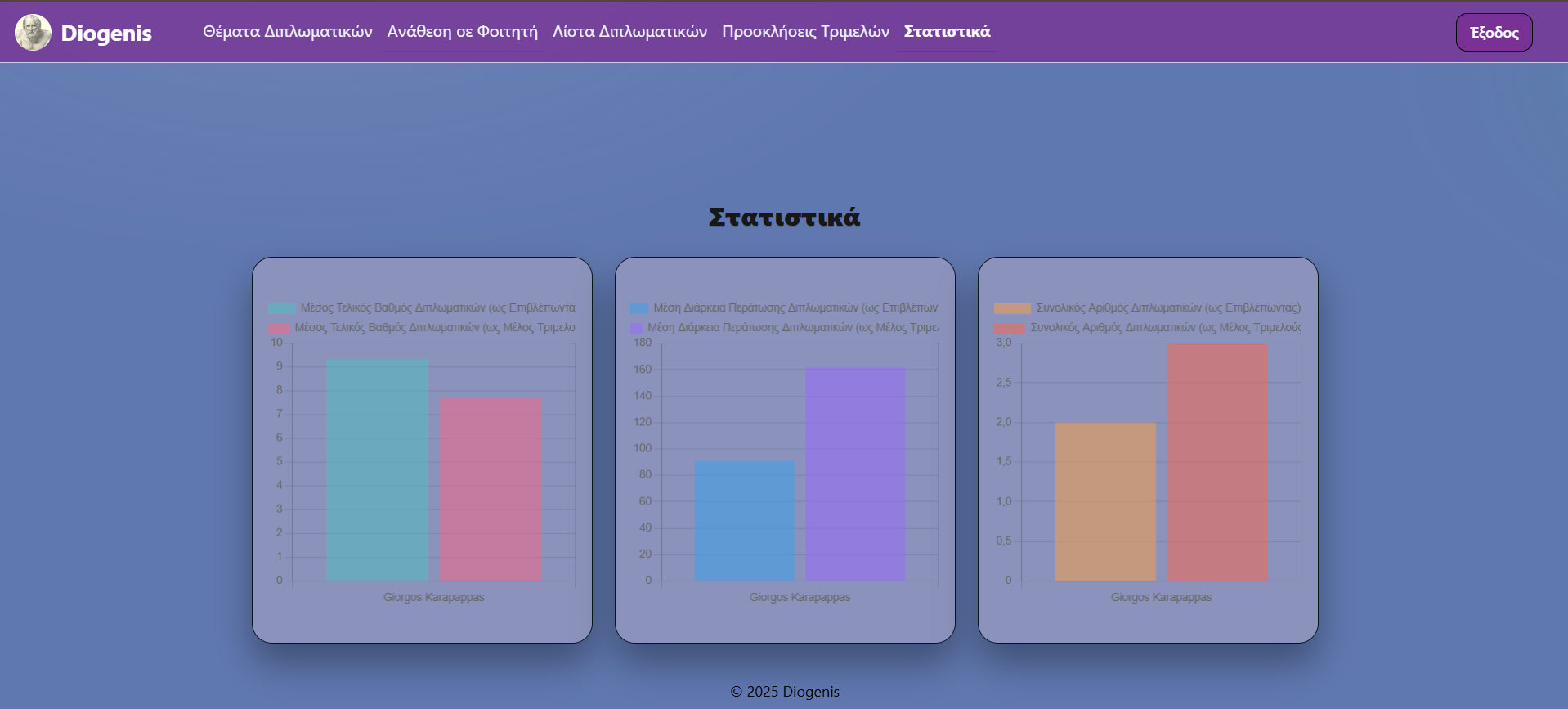


* **Αποδοχή / Απόρριψη Πρόσκλησης σε Τριμελή**

Βλέπει λίστα ενεργών προσκλήσεων, αν υπάρχουν, και τις αποδέχεται ή απορρίπτει.





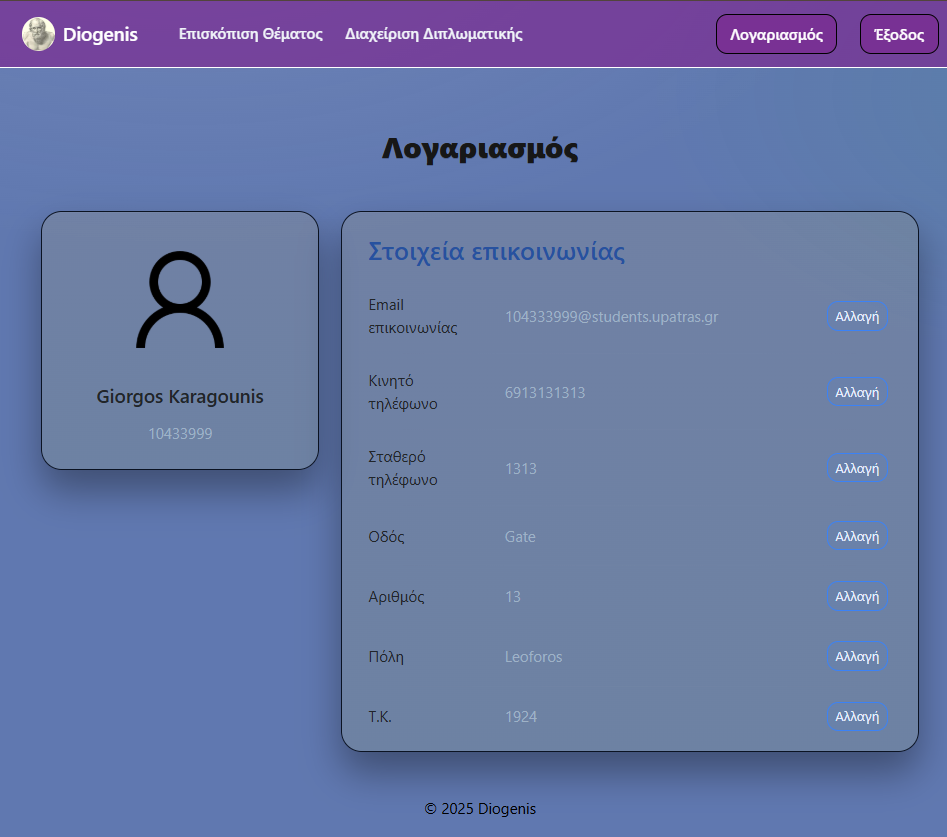
* **Προβολή Στατιστικών**

Βλέπει γραφήματα για: μέσο χρόνο περάτωσης, μέσο βαθμό, πλήθος διπλωματικών (ανάλογα με τον ρόλο του).

**Φοιτητές**

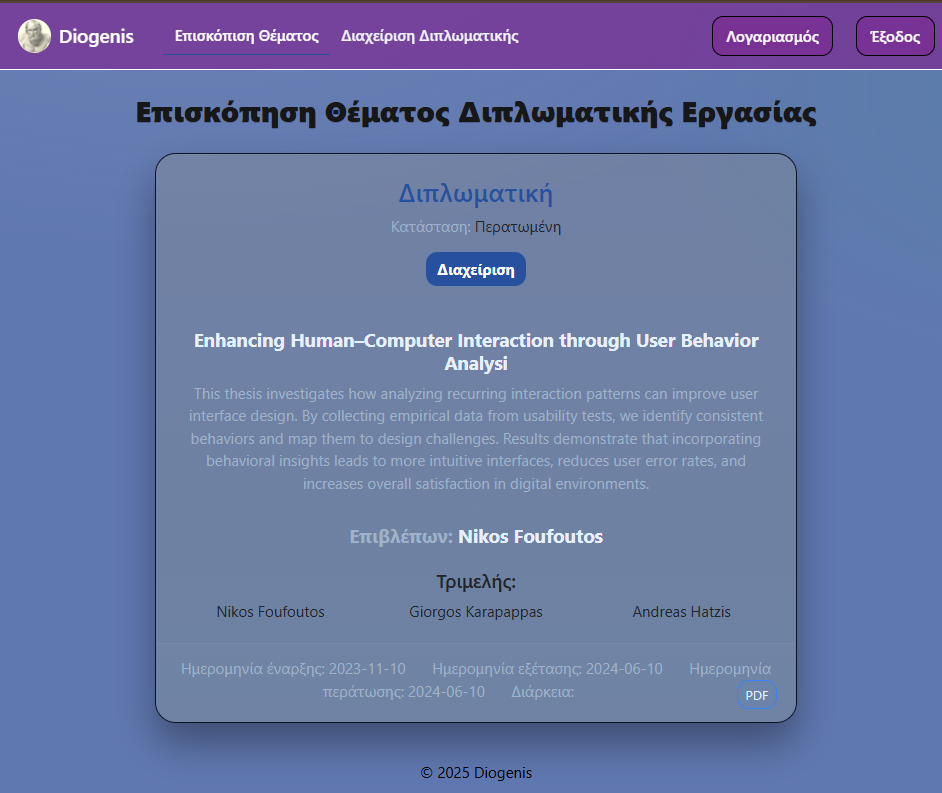
* **Επεξεργασία Λογαριασμού**

Διαχειρίζεται προσωπικά στοιχεία επικοινωνίας.



* **Προβολή Θέματος**

Βλέπει λεπτομέρειες διπλωματικής (θέμα, περιγραφή, κατάσταση, μέλη τριμελούς, χρόνος από ανάθεση).



* **Διαχείριση ΔΕ (ανάλογα με κατάσταση)**
* **Υπό Ανάθεση**: Επιλέγει Διδάσκοντες για τριμελή. Όταν 2 αποδεχτούν, η ΔΕ γίνεται «Ενεργή».
* **Υπό Εξέταση**: Αναρτά πρόχειρο κείμενο, συνδέσμους υλικού, στοιχεία παρουσίασης. Μετά τους βαθμούς, βλέπει πρακτικό εξέτασης και καταχωρεί σύνδεσμο προς Νημερτή.
* **Περατωμένη**: Βλέπει μόνο ιστορικό, πληροφορίες, πρακτικό.

**Γραμματεία**

* **UC-16: Προβολή Ενεργών/Υπό Εξέταση ΔΕ**

**Περιγραφή**: Βλέπει λίστα ΔΕ με λεπτομέρειες, κατάσταση, μέλη τριμελούς, χρόνος από ανάθεση.

* **UC-17: Εισαγωγή Δεδομένων**

**Περιγραφή**: Εισάγει JSON με προσωπικές πληροφορίες χρηστών.

* **UC-18: Διαχείριση ΔΕ**
* **Ενεργή**: Καταχωρεί αριθμό ΓΣ, μπορεί να ακυρώσει ανάθεση (καταχωρεί λόγο).
* **Υπό Εξέταση**: Όταν υπάρχουν βαθμοί και σύνδεσμος Νημερτή, αλλάζει κατάσταση σε «Περατωμένη».

**Δημόσιο Endpoint**

* **UC-19: Προβολή Ανακοινώσεων ΔΕ (χωρίς login)**

**Actor**: Οποιοσδήποτε

**Περιγραφή**: Μέσω δημόσιου endpoint βλέπει ανακοινώσεις παρουσιάσεων για συγκεκριμένο χρονικό εύρος.  
**Παράμετροι**: Επιλογή format JSON ή XML.

# 5. Διαχείριση Cache

[INSERT SCREENSHOT ΑΠΟ ΤΟ SERVER.JS]

Η διπλωματική διαρκεί πολύ χρόνο, άρα τα δεδομένα στο σύστημα δεν αλλάζουν κάθε μέρα. Δεν είναι αποδοτικό να στέλνουμε σε κάθε αίτημα όλα τα δεδομένα, γι’ αυτό ελέγχουμε αν έχουν αλλάξει.

Στον κώδικα **server.js**:

* Χρησιμοποιούμε **HTTP ETag** στα GET. Ο server στέλνει μαζί με την απόκριση ένα ETag (μοναδικό αναγνωριστικό της τρέχουσας κατάστασης των δεδομένων). Στα επόμενα αιτήματα ο client στέλνει την τιμή με If-None-Match. Αν τα δεδομένα δεν έχουν αλλάξει, ο server απαντά με 304 Not Modified χωρίς να ξαναστείλει τα δεδομένα. (υλοποίηση: Express ETag μηχανισμός – server.js γραμμές όπου ρυθμίζεται το app.use(express.static(..., { etag: true }))).
  + Μειώνει τη μεταφορά δεδομένων (bandwidth) όταν το περιεχόμενο είναι αμετάβλητο.
  + Βελτιώνει την απόδοση και ταχύτητα, αφού ο client δεν κατεβάζει όλη την απάντηση ξανά.
  + Παρέχει πιο ακριβή έλεγχο από απλό timestamp επειδή το ETag μπορεί να βασίζεται στο περιεχόμενο ή hash.
* Για τη σελίδα **login**, ορίζονται headers Cache-Control: no-store και Pragma: no-cache, ώστε η σελίδα να μην αποθηκεύεται στον browser. Αυτό εμποδίζει τον χρήστη να γυρίσει πίσω χωρίς logout. (φαίνεται σε server.js στην route του login).
  + Επιτρέπει cache για ένα 24ωρο, οπότε ο client/ο browser μπορεί να εξυπηρετεί τις ίδιες σελίδες χωρίς να κάνει αίτημα στον server κάθε φορά.
  + Μειώνει την καθυστέρηση (“latency”) και το φορτίο του server.
  + Εναρμονίζεται με οδηγίες καλών πρακτικών όπου στατικά assets ή πληροφορίες που αλλάζουν σπάνια «κρατιούνται» στην cache για λογικά χρονικά διαστήματα.
* Για στατικές σελίδες όπως **index** και **announcements**, έχουμε ορίσει Cache-Control με maxAge: '1d'. Έτσι φυλάγονται για 1 μέρα στην cache του browser, μειώνοντας τα περιττά αιτήματα και βελτιώνοντας την ταχύτητα. (φαίνεται στο server.js όπου γίνεται app.use(express.static(path.join(\_\_dirname, '../frontend'), { maxAge: '1d' }))).
  + Αποτρέπει αποθήκευση ευαίσθητων/προσωπικών δεδομένων στο cache του browser ή ενδιάμεσες cache.
  + Προστατεύει από το να μπει κάποιος με “back” χωρίς logout.
  + Εναρμονίζεται με ασφαλείς πρακτικές.