

# Ευφυείς Πράκτορες Απαλλακτική Εργασία 2025

## Περιγραφή

Σκοπός της εργασίας είναι η ανάπτυξη ενός **ευφυούς πράκτορα** ή ενός **πολυπρακτορικού συστήματος** που θα χαρακτηρίζεται από συνεργασία πολλών πρακτόρων στα πλαίσια μιας δομημένης ροής εργασίας (agentic workflow) και την αποτελεσματική επικοινωνία μεταξύ των πρακτόρων, με στόχο την επίλυση σύνθετων προβλημάτων ή την παροχή ολοκληρωμένων υπηρεσιών. Το σύστημα θα βασίζεται στη χρήση **Μεγάλων Γλωσσικών Μοντέλων (LLMs)** και συναφών τεχνολογιών. Είναι απαραίτητο να υπάρχει αλληλεπίδραση του συστήματος με τον χρήστη (τροφοδότηση εισόδου/ερωτημάτων, επιστροφή αποτελέσματος). Δεν είναι απαραίτητο ο τρόπος αλληλεπίδρασης με τον πράκτορα ή σύστημα πρακτόρων να είναι μέσω chat (π.χ. μπορεί η είσοδος του χρήστη αν ενσωματώνεται σε προκαθορισμένα prompts).

## Θέμα

Το θέμα της εργασίας είναι **ελεύθερο**. Μπορείτε να χρησιμοποιήσετε διάφορα εργαλεία και frameworks, τεχνικές RAG (Retrieval-Augmented Generation), Function Calling με tools που θα υλοποιήσετε οι ίδιοι, στοιχεία μνήμης για πράκτορες (state) κ.λπ. Δεν είναι απαραίτητη η υλοποίηση γραφικού περιβάλλοντος διεπαφής χρήστη (GUI), αλλά σε περίπτωση που υφίσταται θα εκτιμηθεί θετικά. Η πολυπλοκότητα του τελικού συστήματος θα πρέπει να αντικατοπτρίζει το μέγεθος της ομάδας εργασίας (τουλάχιστον ένα εργαλείο ή ένας πράκτορας για κάθε μέλος).

Για ιδέες μπορείτε να επισκεφθείτε: <https://rdi.berkeley.edu/llm-agents-hackathon/> και <https://lablab.ai/event/autonomous-agents-hackathon> **Προσοχή:** Οι σύνδεσμοι αυτοί παρέχονται αποκλειστικά για έμπνευση. Αντιγραφή ή υποβολή εργασιών που είναι αυτούσιες ή ελαφρώς τροποποιημένες εκδοχές των παραδειγμάτων που παρουσιάζονται σε αυτές τις σελίδες δεν θα γίνει αποδεκτή και θα απορρίπτεται.

Ανοιχτή λίστα με κάποια ενδεικτικά frameworks που μπορείτε να χρησιμοποιήσετε για την υλοποίηση της εργασίας (για διευκόλυνσή σας καθώς παρέχουν έτοιμα υποσυστήματα που χειρίζονται την επεξεργασία του output των μοντέλων για function calling, διαφορετικές τεχνικές prompting, διατήρηση μνήμης/state, κλπ):

- [Llamaindex](#)
- [Smolagents](#)
- [LangChain/LangGraph](#)
- [CrewAI](#)

Επίσης, δεν υπάρχει περιορισμός ως προς το μοντέλο LLM που θα αποτελεί τον “εγκέφαλο” για τους πράκτορες που θα υλοποιήσετε, παρ’όλα αυτά ένας τρόπος να πειραματιστείτε με αρκετές επιλογές είναι το OpenRouter API (<https://openrouter.ai/models>).

## Ενδεικτικά Θέματα

- Προσωπικός βοηθός
- Πράκτορας για εξυπηρέτηση πελατών
- Πράκτορας για συμμόρφωση με κανονισμούς (π.χ. GDPR)
- Βοηθός προγραμματιστή (coding assistant)
- Αναλυτής δεδομένων ή εγγράφων με RAG

## Ομάδες

- Η εργασία είναι **ομαδική** (2-4 άτομα)
- Σε περιπτώσεις που κάποιος/κάποια είναι από μεγαλύτερο έτος και έχει πρόβλημα να ενταχθεί σε κάποια ομάδα, υπάρχει η δυνατότητα για **ατομική εργασία**, κατόπιν ενημέρωσης του διδάσκοντα με [email](#).

## Παραδοτέα

1. **Κώδικας σε public GitHub repo ή zip αρχείο**
2. **Τεχνική αναφορά (report ~3-6 σελίδες)**
  - Περιγραφή συστήματος
  - Τεχνολογίες και αρχιτεκτονική
  - Διαγράμματα ροής
  - Προβλήματα που αντιμετωπίστηκαν
  - Αξιολόγηση/δοκιμή με διαφορετικά user inputs
3. **Βίντεο παρουσίασης της εργασίας προφορικά (5–10 λεπτά)**
  - Επίδειξη του πράκτορα (demo)
  - Επεξήγηση αρχιτεκτονικής και ροής

Σε περίπτωση που τα παραδοτέα είναι μεγάλου όγκου και δεν μπορούν να ανέβουν στο eclass (σε έναν φάκελο zip), παρακαλώ χρησιμοποιήστε (μεριμνώντας για τα δικαιώματα πρόσβασης) κάποιο cloud drive ή repository στο GitHub και ανεβάστε τον ανάλογο σύνδεσμο.