# WhyBother Project – Τελική Αναφορά

## Στόχος

Η εφαρμογή WhyBother Football Stats αναπτύχθηκε ως μέρος της προγραμματιστικής εργασίας του μαθήματος "Προχωρημένα Θέματα Τεχνολογίας και Εφαρμογών Βάσεων Δεδομένων". Κύριος στόχος του έργου ήταν η κατασκευή ενός πλήρως λειτουργικού πληροφοριακού συστήματος το οποίο ενοποιεί δεδομένα από πολλαπλές πηγές, τα κανονικοποιεί και τα παρέχει μέσω ενός API για χρήση σε διαδραστικό περιβάλλον οπτικοποίησης.

## Αρχιτεκτονική

Η αρχιτεκτονική της εφαρμογής είναι τριεπίπεδη, αποτελούμενη από ένα ETL pipeline σε Python, ένα backend με FastAPI και SQLAlchemy, και ένα frontend περιβάλλον βασισμένο σε React και Next.js. Η pipeline επεξεργάζεται αρχεία CSV με δεδομένα χωρών, αγώνων, γκολ, πέναλτι και ιστορικών ονομάτων και τα φορτώνει σε μία κανονικοποιημένη σχεσιακή βάση δεδομένων MySQL. Κατά τη διαδικασία αυτή εφαρμόζεται ενοποίηση, κανονικοποίηση, αντιμετώπιση aliases, αντιστοιχίσεις πρώην ονομάτων και παράλειψη ελλιπών ή μη έγκυρων εγγραφών, με στόχο την πληρότητα και την ακρίβεια της τελικής βάσης.

Ο backend αναπτύχθηκε με FastAPI και SQLAlchemy, προσφέροντας RESTful API endpoints για ερωτήματα σχετικά με γενικά στατιστικά, προφίλ παικτών, προφίλ χωρών και στατιστικά ανά έτος. Παράλληλα, υποστηρίζεται και δυναμική παραμετροποίηση των queries με χρονικά φίλτρα ή ερωτήματα κατά διοργάνωση. Όλα τα endpoints είναι βελτιστοποιημένα για επικοινωνία με το frontend και σχεδιάστηκαν με γνώμονα την αποδοτικότητα και την επεκτασιμότητα.

Το frontend υλοποιήθηκε με React και Next.js, παρέχοντας πλήρως διαδραστική εμπειρία χρήστη με τη χρήση γραφημάτων (BarChart, LineChart, ScatterPlot) μέσω της βιβλιοθήκης Recharts. Ο χρήστης μπορεί να μεταβεί σε προφίλ χωρών και παικτών, να φιλτράρει αποτελέσματα κατά έτος ή διοργάνωση και να δει συνολικά στατιστικά. Η εφαρμογή ανταποκρίνεται δυναμικά στις αλλαγές του χρήστη, επιτρέποντας άμεση επικοινωνία με το backend μέσω API κλήσεων.

## Υλοποίηση ανάπτυξη έργου και προβλήματα

Η ανάπτυξη του έργου πραγματοποιήθηκε με χρήση MySQL Workbench για τη διαχείριση της βάσης δεδομένων. Αν και υπήρχε αρχικά πρόβλεψη για χρήση Docker και Docker Compose, τεχνικοί περιορισμοί στο σύστημα ανάπτυξης (RAM, αποθηκευτικός χώρος) οδήγησαν στην υλοποίηση του έργου τοπικά, χωρίς πλήρες containerization. Ωστόσο, η αρχιτεκτονική του έργου σχεδιάστηκε ώστε να είναι επεκτάσιμη και έτοιμη για μελλοντική μεταφορά σε περιβάλλοντα containers ή cloud.

Κατά τη διάρκεια του έργου προέκυψαν κρίσιμα προβλήματα που επηρέασαν την υλοποίηση. Σημαντικό σφάλμα αποτέλεσε η υλοποίηση βασικής υπολογιστικής λογικής (όπως υπολογισμοί νικών ή γκολ) σε Python αντί για SQL. Αυτό οδήγησε σε αυξημένο υπολογιστικό φορτίο στον backend, μειωμένη επαναχρησιμοποίηση του κώδικα και παραβίαση της αρχής SQL-first. Επιπλέον, η καθυστερημένη ενσωμάτωση του Docker δημιούργησε ασυνέπειες στο περιβάλλον, καθυστερώντας τη ρύθμιση της βάσης και των υπηρεσιών.

Στο τελικό στάδιο ενσωματώθηκαν SQL views (όπως match_results, country_stats και top_scorers), τα οποία μετέφεραν μεγάλο μέρος της λογικής στο επίπεδο της βάσης δεδομένων. Η χρήση των views απλοποίησε τα API routes, ενίσχυσε την απόδοση και εξασφάλισε μεγαλύτερη συνέπεια στα αποτελέσματα. Ωστόσο, λόγω χρονικών περιορισμών, αρκετά endpoints παρέμειναν εξαρτώμενα από λογική σε Python.

**Τελικές σκέψης**

Η εμπειρία από την ανάπτυξη του WhyBother ανέδειξε σημαντικά διδάγματα. Ο SQL-based σχεδιασμός της βάσης και των views πρέπει να προηγείται της ανάπτυξης του backend για καλύτερο διαχωρισμό αρμοδιοτήτων και βελτιωμένη απόδοση. Η χρήση Docker θεωρείται αναγκαία για σταθερότητα, συμβατότητα και ευκολία στη συντήρηση. Παρά τις τεχνικές προκλήσεις, το έργο ολοκληρώθηκε με επιτυχία: η ETL pipeline ενσωμάτωσε και καθάρισε τα δεδομένα, το REST API ανταποκρίθηκε στα ερωτήματα με αξιοπιστία και το frontend απέδωσε τα στατιστικά σε εύχρηστο διαδραστικό περιβάλλον. Με περισσότερο χρόνο, η εφαρμογή θα μπορούσε να υλοποιηθεί πλήρως με SQL views, βελτιώνοντας ακόμη περισσότερο την απόδοση, τη συντηρητικότητα και την ποιότητα των αποτελεσμάτων.