# HM Markets - Back-end Exercise
# Κωνσταντίνος Θωμάς

## Επισκόπηση

Η ανάπτυξη αυτής της άσκησης έγινε χρησιμοποιώντας **απλή PHP**, δίνοντας προτεραιότητα στην αποφυγή υπερβολικού κώδικα και αυτοματοποιημένων διαδικασιών που ενδέχεται να εισάγουν κενά ασφαλείας. Αυτή η προσέγγιση οδήγησε στην απόφαση να μην χρησιμοποιηθούν frameworks όπως το Laravel και το Drupal, τα οποία συνήθως χρησιμοποιώ. Για το front-end, αξιοποίησα κάποια προκατασκευασμένα εργαλεία επικύρωσης από το Bootstrap για την αυτόματη εμφάνιση σφαλμάτων.

---

## Χρησιμοποιούμενες Τεχνολογίες

- **Απλή PHP**
- **Απλή JavaScript**
- **Bootstrap 5.3** (CSS Framework)

**Τοπικό περιβάλλον διακομιστή:** XAMPP v3.3.0 (PHP 8.2, Apache)

---

## Δομή Έργου

Το έργο περιλαμβάνει τα ακόλουθα αρχεία:

- register-front.php (Φόρμα εγγραφής front-end)
- style.css (Προσαρμοσμένο CSS)
- bg.png (Εικόνα φόντου)
- HFM.png (Εικόνα λογοτύπου)
- register-back.php (Λογική back-end για την εγγραφή)
- Validator.php (Προσαρμοσμένη κλάση επικύρωσης)
- DB.php (Κλάση αλληλεπίδρασης με τη βάση δεδομένων)
- readme.pdf (Αυτή η τεκμηρίωση σε μορφή PDF)

---

## Υλοποίηση Front-End

Το front-end χρησιμοποιεί το **Bootstrap** για τη μορφοποίηση, συμπληρωμένο με ελάχιστο προσαρμοσμένο CSS και δύο εικόνες που παρέχονται στο Figma design.

**Μηνύματα Σφάλματος και Επιτυχίας:** Τα μηνύματα σφάλματος και επιτυχίας, συμπεριλαμβανομένων των κόκκινων περιγραμμάτων για τα μη έγκυρα πεδία εισαγωγής, ακολουθούν την προτεινόμενη λογική εμφάνισης του Bootstrap.

---

# Μέτρα Ασφαλείας

## CSRF Token

Συμμορφούμενη με τις θεμελιώδεις πρακτικές ασφαλείας, η φόρμα περιλαμβάνει ένα **CSRF token**. Ενώ πρωτίστως προστατεύει τους συνδεδεμένους χρήστες από ακούσιες ενέργειες (π.χ. μη εξουσιοδοτημένες χρηματικές μεταφορές σε έναν κακόβουλο κλώνο ιστότοπου), αποτελεί μια τυπική πρακτική για όλες τις φόρμες. Αν και ο κίνδυνος για μια φόρμα εγγραφής είναι μικρότερος, η εφαρμογή προστασίας CSRF είναι ζωτικής σημασίας για τη διατήρηση της συνολικής ασφάλειας της εφαρμογής.

## Επικύρωση Front-End

Η επικύρωση του front-end για τα πεδία **First Name**, **Last Name** και **Email** χρησιμοποιεί τα ενσωματωμένα εργαλεία του Bootstrap. Αυτή η επικύρωση βελτιώνει κυρίως την εμπειρία του χρήστη παρέχοντας άμεση ανατροφοδότηση και μειώνει την περιττή επεξεργασία από το back-end. Δεν αποτελεί υποκατάστατο της ισχυρής επικύρωσης του back-end.

---

# Σκεπτικό Υλοποίησης και Επιλογές Σχεδιασμού

## Πεδία Επιλογής (Χώρα, Εμπειρία)

Για τα πεδία επιλογής **Χώρα** και **Εμπειρία**, επέλεξα να χρησιμοποιήσω **αριθμητικές τιμές** από τη φόρμα προς τη βάση δεδομένων. Τα πλεονεκτήματα αυτής της προσέγγισης περιλαμβάνουν:

- Ταχύτερη μεταφορά δεδομένων.
- Μειωμένο χώρο αποθήκευσης στη βάση δεδομένων.
- Ευκολότερη τροποποίηση των λιστών επιλογών.

Σε μελλοντικές επαναλήψεις, εάν χρειαστούν περισσότερες επιλογές, η ανάκτηση των επιλογών ως ζεύγη (id, label) από τη βάση δεδομένων θα αποτελούσε μια πιο επεκτάσιμη λύση. Αυτή η επιλογή σχεδιασμού προβλέπει μελλοντικές αλλαγές, απλοποιώντας τη συντήρηση.

## Πεδίο Κωδικός Κλήσης Τηλεφώνου

Το πεδίο του κωδικού κλήσης τηλεφώνου **συμπληρώνεται αυτόματα βάσει της επιλεγμένης χώρας** και δεν είναι επεξεργάσιμο από τον χρήστη. Αυτό συμβαίνει επειδή μόλις επιλεγεί μια χώρα, ο κωδικός κλήσης καθορίζεται. Σε ένα σενάριο όπου το πεδίο "Χώρα" απουσιάζει, το πεδίο του κωδικού κλήσης θα ήταν μια επιλέξιμη επιλογή. Αυτή η προσέγγιση ελαχιστοποιεί το σφάλμα του χρήστη και βελτιστοποιεί τη διαδικασία εγγραφής.

# Υλοποίηση Back-End

Το αρχείο register-back.php χειρίζεται την κύρια λογική για την επεξεργασία των αιτημάτων εγγραφής.

## Βήματα Υλοποίησης

### 1. Επικύρωση Μεθόδου Αίτησης

Το πρώτο βήμα είναι να επαληθευτεί ότι η μέθοδος του αιτήματος είναι **"POST"**. Εάν χρησιμοποιηθεί διαφορετική μέθοδος, επιστρέφεται ένα γενικό μήνυμα σφάλματος. Αυτό αποτρέπει πιθανές ευπάθειες ασφαλείας, καθώς δεν παρέχει συγκεκριμένες λεπτομέρειες σε κακόβουλες προσπάθειες που ενδέχεται να αναζητούν αδυναμίες.

### 2. Επικύρωση CSRF Token

Το CSRF token επικυρώνεται (όπως αναλύθηκε παραπάνω). Επιστρέφεται ένα γενικό μήνυμα σφάλματος για μη έγκυρα tokens, και πάλι για να αποφευχθεί η αποκάλυψη τεχνικών λεπτομερειών σε πιθανούς επιτιθέμενους.

### 3. Επικύρωση Πεδίων

Η επικύρωση πεδίων ακολουθεί μια ευέλικτη, εμπνευσμένη από το Laravel προσέγγιση, χρησιμοποιώντας έναν πίνακα κανόνων με σαφή, προγραμματιστικά ονόματα για κάθε πεδίο. Αυτός ο σχεδιασμός επιτρέπει την εύκολη τροποποίηση και επέκταση. Η προσθήκη ενός νέου κανόνα επικύρωσης για ένα πεδίο συχνά απαιτεί μόνο μία γραμμή κώδικα.

Η προσαρμοσμένη κλάση **Validator** αναπτύχθηκε με την ίδια φιλοσοφία. Η προσθήκη ενός νέου κανόνα επικύρωσης είναι τόσο απλή όσο η προσθήκη μιας ακόμη case στην λογική της.

Ένας αξιοσημείωτος κανόνας είναι ο sanitized_text. Ο σκοπός του είναι να αποτρέψει την εισαγωγή κακόβουλων συμβολοσειρών όπως <script> alert('hacked'); </script>. Ενώ τα frameworks συχνά χειρίζονται αυτόματα κατά την λήψη αιτημάτων και την απόδοση μεταβλητών, ο ρητός καθαρισμός συμβολοσειρών πριν την αποθήκευσή τους στη βάση δεδομένων είναι μια καλή γενική πρακτική, ακόμα και όταν ειδικοί χαρακτήρες ενδέχεται να αποθηκευτούν σκοπίμως (π.χ. από έναν επεξεργαστή εμπλουτισμένου κειμένου).

### 4. Καταχώρηση στη Βάση Δεδομένων

Τα δεδομένα αποθηκεύονται σε μια απλή βάση δεδομένων **SQLite**, η οποία είναι ιδανική για τις ανάγκες αυτής της άσκησης. Δύο βασικές πτυχές αξίζει να σημειωθούν:

1. **Πρόληψη SQL Injection:** Για την πρόληψη επιθέσεων SQL injection, όλες οι λειτουργίες της βάσης δεδομένων χρησιμοποιούν τις μεθόδους **prepare() και bindParam() του PDO**. Η άμεση εκτέλεση παρεμβαλλόμενων ερωτημάτων (π.χ. $pdo->exec($sql)) αποφεύγεται αυστηρά, καθώς εκθέτει την εφαρμογή σε σοβαρές ευπάθειες.
2. **Διαχείριση Σφαλμάτων:** Εάν η αποθήκευση στη βάση δεδομένων αποτύχει, ο χρήστης **δεν λαμβάνει λεπτομερή τεχνικά μηνύματα σφάλματος**. Τα μηνύματα που

επιστρέφονται έχουν σχεδιαστεί για να βοηθήσουν τον χρήστη να διορθώσει την εισαγωγή του, και όχι να εκθέσουν εσωτερικά χαρακτηριστικά της εφαρμογής ή τεχνικές βλάβες. Η παροχή υπερβολικών πληροφοριών μπορεί να μπερδέψει τον χρήστη και να βοηθήσει πιθανούς επιτιθέμενους.