ΥΠΗΡΕΣΙΕΣ ΣΤΟ ΥΠΟΛΟΓΙΣΤΙΚΟ ΝΕΦΟΣ ΚΑΙ ΤΗΝ ΟΜΙΧΛΗ (ΠΛΗ 518)

Σπυριδάκης Χρήστος ΑΜ: 2014030022

Για την δημιουργία του web app χρησιμοποιήθηκαν για την βάση το image **mysql:5.7.13**, για το phpmyadmin το **phpmyadmin/phpmyadmin:4.8** ενώ για το web το **php:7.2.6-apache**, το οποίο περιλαμβάνει τόσο τον Apache Server οσο και την php εγκατεστημένη σε αυτόν. Το μόνο που χρειάστηκε ως έξτρα είναι να εγκατασταθεί το mysqli extension προκειμένου να έχουμε πρόσβαση στην βάση από την php.

Το πρότζεκτ μπορεί να τρέξει σε οποιοδήποτε περιβάλλον είναι εγκατεστημένο το docker v19.03.4 or later (δοκιμάστηκε σε αυτό) και το docker-compose v1.24.0 με την χρήση της εντολής **docker-compose up** και σε αυτό το url http://localhost:8080/ (port: 8081 για το phpmyadmin) θα έχουμε πρόσβαση στην εφαρμογή. Κατά την δημιουργία των containers αρχικοποιείται η βάση με το αρχείο .sql που υπάρχει στο directory mysql/ αυτόματα, κάνοντας ουσιαστικά bind mount (όπως αναφέρεται στο official documentation της mysql στο docker hub ως ένας από τους δυνατούς τρόπους Initializing a fresh instance) το φάκελο που περιλαμβάνει το .sql με το /docker-entrypoint-initdb.d. Σε αυτό περιλαμβάνονται μερικοί Teachers και Students ήδη. Ένας εξ' αυτών είναι ο Teacher με Username: **JDoe_1** και Password: **johnPass1**. Τα secret keys της βάσης υπάρχουν στο .env αρχείο προκειμένου να μπορούν εύκολα τόσο να παραμετροποιηθούν όσο και να μην είναι προσβάσιμα σε μελλοντικό deployment.

Σε οποια form χρειάζονται γίνεται έλεγχος ορθότητας των δεδομένων εισόδου τόσο στο frontend - με την χρήση attributes της html (π.χ. maxlength, required, pattern) - οπως επίσης και στο backend ως διπλή επαλήθευση για να σιγουρευτούμε ότι έχουν το περιεχόμενο που πρέπει. Παρόλα αυτά, προκειμένου να σιγουρευτούμε ότι ακόμα και αν ένα πεδίο εισόδου έχει την επιτρεπτή μορφή (π.χ. το όνομα πατέρα έχει λογική να περιλαμβάνει ΜΟΝΟ χαρακτήρες) σε κάθε επικοινωνία με την βάση χρησιμοποιούμε prepare statements ώστε να μειώσουμε τις πιθανότητες για επιθέσεις (π.χ. sql injection).

Στον φάκελο includes περιλαμβάνονται αρχεία τα οποία μόνη δουλειά έχουν την απάντηση ερωτημάτων ή ενεργειών στο backend. Παραδείγματος χάρη αν είμαστε στην σελίδα DeleteStudent.php μολις πατήσουμε διαγραφή ενός χρήστη μεταβαίνουμε στο deletestudent.inc.php όπου γίνεται η επικοινωνία με την βάση, διαγράφεται αν μπορεί ο Student και επανερχόμαστε στην σελίδα που έκανε την κλήση.

Όλες οι επικοινωνίες από τον client προς το server (Add, Edit, Delete, Login, Register) γινονται με POST μεθοδο προκειμένου να μην είναι εμφανή ευαίσθητες πληροφορίες στο URL, εξαίρεση είναι το Search. Κατά το login δημιουργούμε session με τον server. Όλες οι σελίδες που χρειάζεται να έχουμε πρόσβαση, κάνουν require το header.inc.php το οποίο σιγουρεύεται ότι έχουμε ενεργό session, σε αντίθετη περίπτωση γίνεται redirect στο index.php. Για να είμαστε σίγουροι επίσης ότι δεν υπάρχει δυνατότητα να κληθεί με μη αποδεκτό τρόπο ένα inc αρχείο σιγουρευόμαστε ότι αυτό έχει κληθεί πατώντας το ανάλογο submit button σε κάποιο form. Αυτό σημαίνει ότι, μόνο όταν η σελίδα που πρέπει και μόνο όταν κάνει POST σε αυτό το inc αρχείο τότε θα εκτελεστεί.

Ένα από τις μελλοντικές βελτιώσεις της συγκεκριμένης υλοποίησης είναι να δημιουργηθεί ένα πιο σύγχρονο UI με πιο εύχρηστο πιθανόν UX, που λόγο και άλλων υποχρεώσεων τις σχολής και του γεγονότος ότι από μόνα τους αποτελούν επιστημονικούς τομείς δεν έγινε εφικτό να τους δοθεί τόση

προσοχή. Επίσης αυτήν την στιγμή τα password αποθηκεύονται σε plaintext στην βάση. Παρόλο που δεν ζητείται ως requirements, έχει γίνει υλοποίηση - που είναι commented out - για να αποθηκεύονται τα hash τους. Ο λόγος που δεν χρησιμοποιείται είναι διότι δεν γίνεται σωστά το verification αυτή την στιγμή. Άλλη μία λοιπόν μελλοντική βελτίωση έχει να κάνει με την αποσφαλμάτωση της συγκεκριμένης ενέργειας.

BONUS: Σύμφωνα με το επίσημο documentation της mysql προτείνονται δύο τρόποι για την μόνιμη αποθήκευση της βάσης, επιλέχθηκε ο δεύτερος από αυτούς με την χρήση directory. Προκειμένου να μην χάνονται τα δεδομένα της βάσης αν αυτή σταματήσει, έχει προστεθεί στο πεδίο volumes της βάσης στο docker-compose το εξής: ./mysql/data:/var/lib/mysql. Αυτό που καταφέρνουμε γενικά είναι, δημιουργώντας τα container η βάση να αρχικοποιείται με το .sql αρχείο και αμέσως μετά να δημιουργείται ο φάκελος ./mysql/data, στην συνέχεια ότι αλλαγές γίνονται στη βάση αποθηκεύονται στο φάκελο αυτό του project tree μας (για την ακρίβεια αποθηκεύεται όλα τα αρχεία της βάσης). Μπορούμε να επαληθεύσουμε ότι γίνεται μόνιμη αποθήκευση των δεδομένων με το να κάνουμε docker-compose up, να δημιουργήσουμε μερικούς Students και Teachers, να τρέξουμε μετά docker-compose down (by default διαγράφει τα container και τα networks), να ελέγξουμε ότι δεν υπάρχουν volumes στο docker (αν είχαμε χρησιμοποιήσει τον πρώτο τρόπο που προτείνεται δεν θα μπορούσαμε να διαγράψουμε το volume, γιατί τότε θα χάναμε τα δεδομένα) και μετά να ξανα τρέξουμε το docker-compose up. Αν ξανά δούμε τότε τα δεδομένα της database, θα δούμε ότι αυτά που δημιουργήσαμε πριν λίγο δεν έχουν χαθεί και ότι παρόλο το ότι το σύστημα ‘έπεσε’ εξ’ ολοκλήρου αυτά υπάρχουν ακόμα. Το μόνο ‘αρνητικό’ είναι ότι το dir που περιέχει την βάση (./mysql/data) είναι owned by root, επειδή και στο container της mysql ισχύει το ίδιο. Άρα πριν γίνει οποιαδήποτε ενέργεια στο project (π.χ. μεταφορά σε άλλο μηχάνημα) προϋποθέτει να τρέξουμε πρώτα βρισκόμενοι στο root του project tree την εντολή $ sudo chown $USER:$USER ./mysql/data -R.