

ΥΠΗΡΕΣΙΕΣ ΣΤΟ ΥΠΟΛΟΓΙΣΤΙΚΟ ΝΕΦΟΣ ΚΑΙ ΤΗΝ ΟΜΙΧΛΗ

Χρήση Docker για την ανάπτυξη μιας Web εφαρμογής
Κατάρα Σωτηρία Μαρία
2015030040

29/11/2019

Σκοπός της άσκησης είναι η χρήση Docker για την ανάπτυξη μιας Web εφαρμογής. Η εφαρμογή σχεδιάζεται και υλοποιείται ως σύνθεση ανεξάρτητων υπηρεσιών (Services) που επικοινωνούν μεταξύ τους.

1. Διαμόρφωση του περιβάλλοντος της εφαρμογής με χρήση Containers

Όπως ζητήθηκε, η εφαρμογή χρησιμοποιεί τα ακόλουθα εργαλεία: Apache Server, MySQL, και phpMyAdmin. Τα εργαλεία αυτά εγκαθίστανται σε ξεχωριστά containers με τη βοήθεια του Docker. Αφού το port managing των κιβωτίων ρυθμίζεται κατάλληλα, τα κιβώτια μπορούν να επικοινωνούν μεταξύ τους. Όλες οι συνδεσμολογίες υλοποιούνται μέσω του κώδικα που βρίσκεται στο αρχείο `docker-compose.yml`. Αξίζει να σημειωθεί ότι δημιουργήθηκε ένα κιβώτιο "bind mount" στο κιβώτιο του Apache Server ώστε να συνδέεται το directory του κώδικά μας με το root directory του Apache Server.

Στην εφαρμογή αποθηκεύονται στοιχεία για δύο κατηγορίες ανθρώπων, Teachers και Students, των οποίων τα δεδομένα έχουν την σειρά και τους τύπους που δίνονται στην εκφώνηση. Οι πίνακες δημιουργούνται και αρχικοποιούνται μέσω της πλατφόρμας php-MyAdmin.

Για την μόνιμη αποθήκευση δεδομένων στη βάση δημιουργείται χώρος αποθήκευσης (volume) στο κιβώτιο της MySQL ώστε ακόμα και αν το κιβώτιο σταματήσει για κάποιο λόγο να μην χάνονται τα δεδομένα της βάσης.

2. Είσοδος στο σύστημα (login)

Την πραγματοποίηση της εισόδου στο σύστημα και τη δημιουργία της πρώτης σελίδας της εφαρμογής μας, αναλαμβάνει το αρχείο `index.php`. Πατώντας το κουμπί login γίνεται ταυτοποίηση των στοιχείων που εκχωρούνται στα πεδία της φόρμας LOG IN, με τα

στοιχεία που βρίσκονται στον πίνακα Teachers. Εάν είναι σωστά επιτρέπεται η είσοδος σε επόμενη σελίδα.

3. Επεξεργασία στοιχείων στην βάση δεδομένων

Μετά την επιτυχή ταυτοποίηση των στοιχείων του χρήστη και την είσοδο του στο σύστημα γίνεται ανακατεύθυνση στη σελίδα Teacher.php. Σε αυτή την σελίδα, δίνεται η δυνατότητα στον χρήστη να πραγματοποιήσει μία από τις παρακάτω ενέργειες, όπως αυτές παρουσιάζονται στο navigation bar της σελίδας:

- **Add(AddStudent.php)** : Μέσω της επιλογής Add, γίνεται ανακατεύθυνση στην σελίδα AddStudent.php. Εκεί ο χρήστης συμπληρώνει μια φόρμα με τα στοιχεία του φοιτητή που επιθυμεί να εισάγει και πατώντας το κουμπί ADD εκτελείται το ερώτημα insert για προσθήκη του φοιτητή στον πίνακα Students της βάσης.
- **Edit(EditStudent.php)** : Μέσω της επιλογής Edit, γίνεται ανακατεύθυνση στην σελίδα EditStudent.php. Στην σελίδα αυτή εμφανίζεται ο πίνακας με τα στοιχεία των ήδη αποθηκευμένων φοιτητών (ένας φοιτητής ανά σειρά). Στην τελευταία στήλη του πίνακα δίνεται η δυνατότητα επεξεργασίας για κάθε φοιτητή μέσω του συνδέσμου Edit. Πατώντας τον σύνδεσμο αυτό, ο χρήστης ανακατευθύνεται στην σελίδα edit.php όπου συμπληρώνει μια φόρμα αντίστοιχη αυτής στην σελίδα Add. Αφού τα νέα στοιχεία εισαχθούν στην φόρμα, για την ενημέρωση των στοιχείων στην βάση (μέθοδος UPDATE) είναι απαραίτητο το πάτημα του button EDIT από τον χρήστη.
- **Delete(DeleteStudent.php)** : Μέσω της επιλογής Delete, γίνεται ανακατεύθυνση στην σελίδα DeleteStudent.php. Στην σελίδα αυτή εμφανίζεται ο πίνακας με τα στοιχεία των ήδη αποθηκευμένων φοιτητών (ένας φοιτητής ανά σειρά). Στην τελευταία στήλη του πίνακα δίνεται η δυνατότητα διαγραφής για κάθε φοιτητή μέσω του συνδέσμου Delete. Πατώντας τον σύνδεσμο αυτό, η εγγραφή για τον συγκεκριμένο φοιτητή διαγράφεται από τον πίνακα Students, καθώς γίνεται ανακατεύθυνση στην σελίδα delete.php όπου και εκτελείται η μέθοδος DELETE για την συγκεκριμένη εγγραφή.
- **Search(SearchStudent.php)** : Μέσω της επιλογής Search, γίνεται ανακατεύθυνση στην σελίδα SearchStudent.php. Εκεί ο χρήστης συμπληρώνει μια φόρμα με τα στοιχεία id, name και surname του φοιτητή για τον οποίο επιθυμεί να κάνει την αναζήτηση και πατώντας το κουμπί SEARCH γίνεται ανακατεύθυνση στη σελίδα search.php. Στη σελίδα αυτή, τα στοιχεία που αντλούνται από τη συμπλήρωση της φόρμας της προηγούμενης σελίδας (μέθοδος GET), χρησιμοποιούνται ως απαραίτητα κριτήρια στο sql ερώτημα (SELECT) προς τη βάση, με σκοπό την αντιστοίχιση με κάποιον αποθηκευμένο φοιτητή.

Σχόλια-Παρατηρήσεις:

- Η σελίδα session.php δεν επιτρέπει πρόσβαση σε καμία σελίδα χωρίς ταυτοποίηση ακόμη και αν ο χρήστης προσπαθεί να αποκτήσει πρόσβαση απευθείας μέσω του url της σελίδας.
- Μετά την επιτυχή εκτέλεση κάθε λειτουργίας γίνεται ανακατεύθυνση στη σελίδα Teacher.php
- Σε περίπτωση που γίνει απόπειρα προσθήκης ενός φοιτητή με ID που υπάρχει ήδη στη βάση, τότε η εφαρμογή παραμένει στην σελίδα Add_Student.php μέχρι να προστεθούν να ορθά στοιχεία.

- Στο πάνω δεξί μέρος κάθε σελίδας υπάρχει η δυνατότητα log out (logout.php) με ανακατεύθυνση στη σελίδα index.php ενώ αναγράφεται και το ονοματεπώνυμο του χρήστη που έχει κάνει το log in. Αξίζει να σημειωθεί πως κατά το πάτημα των buttons EDIT και ADD, στις αντίστοιχες σελίδες edit.php και Add_Student.php, όπου γίνεται ανακατεύθυνση στην σελίδα Teacher.php, το όνομα και το επώνυμο του login user εξαφανίζονται παρόλο που υπάρχει ο απαραίτητος κώδικας για την εμφάνιση τους.
- Το Docker File χρησιμοποιείται για να συνδέσουμε τον php interpreter με το container της mysql.

Σύνδεση στην Εφαρμογή:

- Url εφαρμογής : <http://127.0.0.1:8000/> με Username: nikoik Password: 123456789
- Url phpMyAdmin : <http://127.0.0.1:8004/> με Username: herausser και Password: herapass

Παραπομπές:

1. <https://www.allphptricks.com/insert-view-edit-and-delete-record-from-database-using-php-and-mysql/>
2. <https://codinginfinite.com/restful-web-services-php-example-php-mysql-source-code/>
3. https://www.tutorialspoint.com/php/php_mysql_login.htm
4. <http://blog.adnansiddiqi.me/create-your-first-php-mysql-application-in-docker/>
5. https://www.youtube.com/watch?v=_mwWxgfZ7Zc