

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑΣ

Κατανεμημένο Σύστημα για Αυτόματη Επικύρωση Κώδικα στο Moodle.

Distributed System for Automatic Code Validation in Moodle

Κηλίκης Μιχαήλ

Επιβλέπων Καθηγητής:

Σαμολαδάς Βασίλειος

Επιτροπή:

Πετράκης Ευριπίδης

Λαγουδάκης Μιχαήλ



Διπλωματική εργασία που υποβλήθηκε στα πλαίσια της ολοκλήρωσης του διπλώματος
Ηλεκτρολόγου Μηχανικού και Μηχανικού Υπολογιστών

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον καθηγητή κ. Σαμολαδά Βασίλειο για την πολύτιμη καθοδήγησή του κατά τη διάρκεια της ανάπτυξης της παρούσας εργασίας, καθώς και για την ευκαιρία που μου έδωσε να διευρύνω τις γνώσεις και τους ορίζοντές μου στον τομέα της ανάπτυξης και της επέκτασης εφαρμογών λογισμικού. Επίσης, θα ήθελα να ευχαριστήσω τον κ. Πετράκη Ευριπίδη και τον κ. Λαγουδάκη Μιχαήλ που δέχτηκαν να είναι μέλη της εξεταστικής επιτροπής.

Τέλος, ευχαριστώ την οικογένειά μου και τους φίλους μου για την υποστήριξη τους κατά την διάρκεια της ανάπτυξης της παρούσας διπλωματικής εργασίας.

Κηλίκης Μιχαήλ
Χανιά, 2022

Πίνακας Περιεχομένων

1. Εισαγωγή	1
1.1 Κίνητρο και Επιστημονική Συνεισφορά	1
1.2. Συστήματα Διαχείρισης Μάθησης	1
1.3. Online Judging Systems.....	2
1.3.1 Αρχιτεκτονική Σχεδίαση	2
1.3.2 Sandboxing	3
2. Παρόμοια Συστήματα	4
2.1 Συστήματα Αυτόματης Βαθμολόγησης	4
2.1.1 Coderbyte	4
2.1.2. CodeRunner	5
2.1 Συστήματα Διαχείρισης Μάθησης	8
2.2.1 Moodle	8
2.2.2 Open eClass	8
3. Χρησιμοποιούμενα Εργαλεία και Τεχνολογίες.....	10
3.1 Frameworks.....	10
3.1.1 Spring-Boot.....	10
3.1.2. ASP.NET	11
3.2. Επικοινωνία εφαρμογών μέσω REST αρχιτεκτονικής.....	15
3.2.1. Διεπαφή API.....	15
3.2.2. RESTful API.....	15
3.3. Docker	15
3.3.1. Πώς λειτουργεί το λογισμικό Docker.....	16
3.3.2. Τα δομικά στοιχεία του Docker	16
3.3.3. Η Αρχιτεκτονική του Docker	17
3.4. Learning Tools Interoperability.....	18
3.4.1. Names and Role Provisioning Services.....	18
3.4.2. Assignment and Grade Services.....	18
3.4.3. Deep Linking	19

4. Περιγραφή Συστήματος	20
4.1. Code-Train	22
4.1.1. Λειτουργίες Code-Train	22
4.1.1. Βάση Δεδομένων Code-Train	26
4.2. Εφαρμογή Sandbox.....	27
4.3 Επικοινωνία Sandbox - Code-Train.....	28
5. Υλοποίηση Συστήματος	29
5.1. Code-Train	29
5.1.1. Δημιουργία Χρηστών και αλληλεπίδραση μέσω LTI	29
5.1.2. Ερωτήσεις και Προφίλ Ερωτήσεων.....	29
5.1.3. Δημιουργία και Εξέταση Διαγωνίσματος.....	30
5.2. Εφαρμογή Sandbox.....	30
6. Αξιολόγηση του Code-Train	31
7. Συμπεράσματα	32
7.1. Περίληψη	32
7.2. Μελλοντική Επέκταση	32
Βιβλιογραφία	33

Εικόνες

Εικόνα 1: Αρχιτεκτονική σχεδίαση Code Runner.....	5
Εικόνα 2: Requests του Jobe Server.....	7
Εικόνα 3: Αλληλεπίδραση Model-View-Controller	12
Εικόνα 4: Αρχιτεκτονική σχεδίαση Docker	17
Εικόνα 5: Υπηρεσίες LTI Advantage	19
Εικόνα 6: Εγγραφή Χρήστη στο Code-Train μέσω Learning Management System	20
Εικόνα 7: Διαδικασία υποβολής και βαθμολόγησης κώδικα	21
Εικόνα 8: Δημιουργία Προφίλ Ερώτησης.....	22
Εικόνα 9: Δημιουργία Ερώτησης.....	23
Εικόνα 10:Επισύναψη JSON αρχείου Διαμόρφωσης.....	23
Εικόνα 11: Δημιουργία Διαγωνίσματος	24
Εικόνα 12: Εξαγωγή Στοιχείων Διαγωνίσματος Από LMS μέσω LTI εργαλείου	24
Εικόνα 13: Εισαγωγή Βαθμολογίας Ερωτήσεων.....	25
Εικόνα 14: Βάση Δεδομένων Code-Train.....	26

Συντομογραφίες

LMS: Learning Management System

MVC: Model View Controller

REST: Representational State Transfer

JSON: JavaScript Object Notation

LTI: Learning Interoperability

API: Application Programming Interface

NRPS: Names and Role Provisioning Services

AGS: Assignment and Grade Services

1. Εισαγωγή

1.1 Κίνητρο και Επιστημονική Συνεισφορά

Η ηλεκτρονική μάθηση χρησιμοποιείται από τα περισσότερα εκπαιδευτικά ιδρύματα ως μέσο διδασκαλίας για τη διευκόλυνση μαθημάτων μεικτής μάθησης και εξ αποστάσεως εκπαίδευσης. Οι κλασικές μέθοδοι διδασκαλίας συνδυάζονται με νέες μεθόδους και ιδέες βασισμένες στην τεχνολογία, επιτρέποντας την ανάπτυξη νέων στρατηγικών εκπαίδευσης. Στα σύγχρονα συστήματα ηλεκτρονικής μάθησης, τα διαδικτυακά εργαλεία επηρεάζουν σημαντικά τον σχεδιασμό των μαθημάτων. Τα Συστήματα Διαχείρισης Μάθησης γίνονται ολοένα και πιο λειτουργικά, προσφέροντας στους εκπαιδευτές πληθώρα εργαλείων για την ανάπτυξη και την εξέλιξη των μεθόδων διδασκαλίας τους. Τέτοια εργαλεία μπορούν να κατασκευαστούν, προκειμένου να επιτευχθούν ανταγωνιστικές συνεργατικές ή ατομικές εκπαιδευτικές δραστηριότητες στα πλαίσια ενός μαθήματος.

Στα πλαίσια της ηλεκτρονικής μάθησης έχει πραγματοποιηθεί σημαντική προσπάθεια για την κατασκευή εργαλείων για την αυτόματη βαθμολόγηση ερωτήσεων προγραμματισμού ή άλλων μορφών εργασιών των μαθητών. Η πιο κοινή τεχνική αυτόματης βαθμολόγησης είναι η αντιστοίχιση λέξεων η φράσεων, οι οποίες τίθενται από τον εκπαιδευτή, και εξάγονται από τον εκπαιδευόμενο. Οι περισσότερες εργασίες στα μαθήματα της τριτοβάθμιας εκπαίδευσης μπορούν να βαθμολογηθούν αυτόματα χρησιμοποιώντας συστήματα διαχείρισης μάθησης, χωρίς δυσκολίες. Ωστόσο στα μαθήματα επιστήμης των υπολογιστών, και ειδικότερα στις εργασίες ή στις εξετάσεις προγραμματισμού, χρειάζεται ένα περεταίρω βήμα για την αυτοματοποιημένη αξιολόγηση των απαντήσεων των φοιτητών.

Η αυτόματη αξιολόγηση προγραμμάτων, πέρα από την ηλεκτρονική μάθηση, εφαρμόζεται και σε άλλους τομείς. Εφαρμόζεται σε διεθνείς διαγωνισμούς προγραμματισμού, για την εξοικονόμηση χρόνου και την διασφάλιση της ακεραιότητας και της αμερόληπτης αξιολόγησης των προγραμμάτων των διαγωνιζομένων, καθώς επίσης και στην αγορά εργασίας, σε συνεντεύξεις υποψηφίων για θέσεις προγραμματιστών.

1.2. Συστήματα Διαχείρισης Μάθησης

Τα συστήματα διαχείρισης μάθησης είναι εφαρμογές λογισμικού για την διαχείριση, αυτοματοποίηση και παρακολούθηση εκπαιδευτικών προγραμμάτων. Αυτά τα συστήματα αποτελούν το μεγαλύτερο τμήμα στην αγορά συστημάτων μάθησης και αντιμετώπισαν τεράστια ανάπτυξη με την εμφάνιση της εξ αποστάσεως εκπαίδευσης κατά την διάρκεια της πανδημίας Covid-19. (Ellis, 2009) Επικεντρώνονται στην παροχή διαδικτυακής μάθησης, αλλά παράλληλα λειτουργούν και ως πλατφόρμες διαδικτυακού περιεχομένου μάθησης. Αφορά εκπαιδευτικούς οι οποίοι αναρτούν το διδακτικό τους υλικό και οργανώνουν τα μαθήματά τους. Οι μαθητές έχουν πρόσβαση στο υλικό αυτό, συμμετέχουν στις εργασίες τους και αξιολογούνται. Παρέχει δυνατότητες επικοινωνίας των μαθητών τόσο μεταξύ τους, όσο και με τους εκπαιδευτικούς σε ένα οργανωμένο περιβάλλον.

Ανάμεσα στα διάφορα συστήματα διαχείρισης μάθησης μπορεί να υπάρχουν σημαντικές διαφοροποιήσεις, αλλά υπάρχουν επίσης πολλά κοινά χαρακτηριστικά. Οι περισσότερες πλατφόρμες περιέχουν κοινές ενότητες όπως ανακοινώσεις, ημερολόγια διαγωνισμάτων ή ημερήσιο πρόγραμμα, βιβλιοθήκες εικόνων και εγγράφων καθώς και ενημερώσεις παρουσίας και αξιολόγηση εκπαιδευόμενων. Στις κοινές δυνατότητες των πλατφόρμων συγκαταλέγονται η αποστολή email μεταξύ εκπαιδευόμενων και εκπαιδευτικών, η διαμοίραση αρχείων και ανάληψη εργασιών, η συμμετοχή σε συζητήσεις, και χαρακτηρίζονται από υψηλό επίπεδο ασφάλειας προσωπικών πληροφοριών.

Κύριο γνώρισμα των πλατφόρμων διαχείρισης μάθησης αποτελούν τα εργαλεία εξυπηρέτησης χρηστών. Οι εκπαιδευτικοί με την χρήση των ανακοινώσεων ενημερώνουν τους μαθητές για σημαντικά γεγονότα που αφορούν τα μαθήματά τους. Επιπρόσθετα, μπορούν να δημιουργήσουν ερωτηματολόγια ή έρευνες με σχετική ευκολία ώστε να μάθουν την γνώμη των μαθητών και να διαμορφώσουν κατάλληλα το μάθημά τους. Οργανώνουν με ασφάλεια τα δεδομένα τους και έχουν πρόσβαση οποιαδήποτε στιγμή με την χρήση των βάσεων δεδομένων. Με το εργαλείο της Παρουσίας (attendance) ελέγχουν την συμμετοχή των χρηστών και με τις Εκθέσεις βλέπουν αναφορές για την πρόοδο και την επίτευξη δεξιοτήτων των μαθητών. Επίσης οι εκπαιδευτικοί έχουν στην διάθεσή τους *εργαλεία* για την δημιουργία διαγωνισμάτων και μπορούν να εκτυπώσουν και να στείλουν με email τα αποτελέσματα. Έχουν τη δυνατότητα να ορίσουν την βαθμολογική κλίμακα κάθε διαγωνίσματος, να δημιουργήσουν ερωτήσεις για το εκάστοτε διαγώνισμα, επιλέγοντας ανάμεσα σε μεγάλο αριθμό τύπων ερωτήσεων, και να δημοσιεύσουν τα αποτελέσματα στο Gradebook. Παράλληλα, τα εργαλεία αυτά αφορούν και τους εκπαιδευόμενους. Έχουν πρόσβαση στα προσωπικά τους στοιχεία μέσα από την σελίδα My Profile. Επιπλέον στο εργαλείο Calendar οι εκπαιδευόμενοι βλέπουν μια λίστα από προσεχή γεγονότα με τις αντίστοιχες ημερομηνίες. Με τα εργαλεία Αυτοαξιολόγησης και Homework οι μαθητές εξασκούν και εφαρμόζουν την γνώση τους μέσω κάποιων online test ή ερωτηματολογίων που έχει θέσει ο καθηγητής, και παράλληλα μπορούν να δημιουργήσουν λίστα με τις υποχρεώσεις τους που εκκρεμούν.

1.3. Online Judging Systems

Οι διαδικτυακοί κριτές, ή Online Judges είναι συστήματα σχεδιασμένα με σκοπό την αξιόπιστη αξιολόγηση κώδικα υποβαλλόμενου από τους χρήστες. Ο κώδικας των χρηστών μεταγλωττίζεται και εκτελείται σε ασφαλές περιβάλλον. Οι Online Judges είναι αρκετά δημοφιλή λόγω της εφαρμογής τους σε πολλούς τομείς, όπως στην εκπαίδευση, σε διαδικασίες πρόσληψης στον τομέα της εργασίας και σε προγραμματιστικούς διαγωνισμούς. Αξίζει να σημειωθεί επίσης πως τα συστήματα αυτά μπορούν να εφαρμοστούν με επιτυχία στην επίλυση σύνθετων βιομηχανικών και επιστημονικών προκλήσεων.

1.3.1 Αρχιτεκτονική Σχεδίαση

Τα συστήματα Online Automated Judges αποτελούνται από δύο τμήματα, το περιβάλλον εκτέλεσης κώδικα sandbox, και το τμήμα Web. Προωθείται η Micro Service Αρχιτεκτονική ώστε τα συστήματα να είναι ελαφριά, εύελικτα, επεκτάσιμα με αποτέλεσμα την επίτευξη βελτιστοποίησης πόρων.

Το Web τμήμα ενός Online Judge παρέχει το γραφικό περιβάλλον στον χρήστη και αποθηκεύει τα όποια δεδομένα προκύπτουν από την χρήση του συστήματος σε βάση δεδομένων. Επίσης είναι υπεύθυνο

για την επικοινωνία του sandbox με οποιαδήποτε επιχειρηματική λογική έχει το σύστημα, όπως για παράδειγμα σύνδεση ή εγγραφή χρηστών.

Τα συστήματα Online Judges αντιμετωπίζουν απειλές από την υποβολή κακόβουλου κώδικα, η οποία μπορεί να προκαλέσει την αποτυχία του διακομιστή ή την συντριβή του σκληρού δίσκου του κεντρικού υπολογιστή. Για αυτόν τον λόγο, κατά τον σχεδιασμό ενός Online Judge συστήματος, πρέπει οι χρήστες να υποβάλλουν τον κώδικά τους σε ασφαλές απομονωμένο μέρος, στο οποίο θα έχουν πρόσβαση μόνο σε περιορισμένους πόρους. Το απομονωμένο αυτό τμήμα ονομάζεται περιβάλλον Sandbox. Το τμήμα Sandbox διαχωρίζεται από το τμήμα Web του συστήματος. Το Sandbox είναι υπεύθυνο για να δέχεται αιτήματα από την πλευρά του Web, να μεταγλωττίζει και να εκτελεί τον κώδικα που υποβάλλεται από τους χρήστες και στη συνέχεια, να εξάγει αποτελέσματα σύμφωνα με τις προδιαγραφές που έχουν τεθεί. Έπειτα επιστρέφει τα αποτελέσματα στο τμήμα Web, και η διαδικασία αυτή επαναλαμβάνεται για κάθε υποβολή κώδικα των χρηστών. (Wasik, 2017)

1.3.2 Sandboxing

Υπάρχουν διάφοροι τρόποι με τους οποίους ένα σύστημα online judge διασφαλίζει την ομαλή και ασφαλή εκτέλεση κώδικα. Η πιο απλή μέθοδος είναι να τρέχει μέσα σε μία εικονική μηχανή με περιορισμούς στη μνήμη και στον επεξεργαστή. Ωστόσο αυτή η μέθοδος δεν είναι και η πλέον αποδοτική.

Η χρήση Docker, δημιουργώντας Container για κάθε εκτέλεση κώδικα απομονώνει το περιβάλλον εκτέλεσης, και παρέχει το δικό του file system και έτσι ο κώδικας που εκτελείται δεν έχει πρόσβαση στο file system του κεντρικού συστήματος. Παρόλη την απομόνωση του συστήματος η μέθοδος αυτή δεν είναι τελείως ασφαλής, και αυτό γιατί τα containers δεν προσφέρουν απομόνωση σε επίπεδο πυρήνα (kernel-level isolation). Έτσι τα containers εξακολουθούν να χρησιμοποιούν τον κεντρικό πυρήνα για τη λειτουργικότητά τους. Οποιαδήποτε κλήση συστήματος γίνεται από την εφαρμογή μέσα στο container, θα εκτελέσει κάποια λειτουργία του κεντρικού πυρήνα. Το πρόβλημα αυτό λύνεται με τη χρήση εφαρμογών Userspace kernel (HN, 2021). Οι εφαρμογές αυτές αποτελούν λογισμικό που τρέχει αποκλειστικά σε user-mode και έχει λιγότερα δικαιώματα. Επίσης προσομοιώνουν την λειτουργία του κεντρικού πυρήνα και δέχονται κλήσεις συστήματος, “κρύβοντας” τον αληθινό και προστατεύοντας το κεντρικό σύστημα.

Τέλος, μία εξαιρετικά αποδοτική μέθοδος για την διασφάλιση της ασφάλειας του sandbox συστήματος είναι η χρήση της εντολής chroot jail. Το chroot είναι μια λειτουργία των unix συστημάτων που αλλάζει το root directory του συστήματος σε αυτό που καθορίζει ο χρήστης. Οποιαδήποτε διεργασία που εκτελείται μετά από μια λειτουργία chroot έχει πρόσβαση μόνο στο καθορισμένο root directory και σε εκείνα που βρίσκονται μέσα σε αυτό.

2. Παρόμοια Συστήματα

2.1 Συστήματα Αυτόματης Βαθμολόγησης

2.1.1 Coderbyte

Πρόκειται για μία πλατφόρμα που αφορά την αξιολόγηση και την εκπαίδευση υποψηφίων για θέσεις προγραμματιστών και μηχανικών λογισμικού. Περιέχει περισσότερα από 300 προβλήματα-ερωτήσεις που προέρχονται από συνεντεύξεις εταιριών με θέματα που αφορούν κυρίως front-end, back-end, δομές δεδομένων και θέματα ανάπτυξης αλγορίθμων. Κάθε χρήστης έχει πρόσβαση στις απαντήσεις των υπόλοιπων χρηστών καθώς επίσης και στις επίσημες προτεινόμενες λύσεις για τις δημοφιλείς ερωτήσεις.

2.1.1.1 Δημιουργία και Διαμόρφωση Προφίλ Ερώτησης

Οι χρήστες του συστήματος διακρίνονται σε δύο κατηγορίες, τους εξεταστές και τους εξεταζόμενους. Οι εξεταστές αρχικά θα πρέπει να δημιουργήσουν λογαριασμό και να συνδεθούν στην πλατφόρμα. Κατά τη δημιουργία μιας αξιολόγησης, μπορούν είτε να δημιουργήσουν, είτε να χρησιμοποιήσουν κάποιο έτοιμο προφίλ αξιολόγησης από την βιβλιοθήκη της πλατφόρμας. Υπάρχουν τρεις τρόποι δημιουργίας προφίλ αξιολόγησης. Η πρώτη επιλογή είναι η δημιουργία κενής αξιολόγησης, με την οποία ο εξεταστής δημιουργεί το προφίλ της αξιολόγησης χωρίς καμία καθοδήγηση από την πλατφόρμα. Επιπλέον υπάρχει η επιλογή της δημιουργίας αξιολόγησης βάσει ρόλου, στην οποία ο εξεταστής καλείται να καθορίσει την διάρκεια της αξιολόγησης, το απαιτούμενο επίπεδο εμπειρίας για την διευθέτησή της, καθώς και την γλώσσα προγραμματισμού που θα χρησιμοποιηθεί. Τέλος υπάρχει το προφίλ βάσει ικανοτήτων, στο οποίο οι εξεταστές επιλέγουν πολλές γλώσσες προγραμματισμού ή τεχνολογίες για την διευθέτησή της αξιολόγησης. Αφού δημιουργηθεί, ή επιλεγεί το προφίλ, ο εξεταστής το διαμορφώνει. Οι μεταβλητές που πρέπει να συμπληρωθούν για την διαμόρφωση του προφίλ είναι η χρονική διάρκεια της αξιολόγησης, ο ελάχιστος βαθμός που απαιτείται για να θεωρηθεί η αξιολόγηση πετυχημένη, και ορισμένες επιλογές που αφορούν τους υποψήφιους, όπως για παράδειγμα αν επιτρέπεται η επανυποβολή καθώς και η προβολή των αποτελεσμάτων κατά την διευθέτηση της αξιολόγησης.

2.1.1.2 Τύποι Ερωτήσεων και αξιολόγηση

Η κάθε αξιολόγηση αποτελείται από τέσσερις διαφορετικούς τύπους ερωτήσεων, τις ερωτήσεις κώδικα, πολλαπλής επιλογής, ανάπτυξης και προσαρμοσμένες εργασίες. Οι πρώτες δύο κατηγορίες βαθμολογούνται αυτόματα. Για τις ερωτήσεις κώδικα, ο εξεταστής επιλέγει από την βιβλιοθήκη της

πλατφόρμας τις ερωτήσεις που θέλει να χρησιμοποιήσει, και η βαθμολογία επιτυγχάνεται βάσει ενός σετ εισόδων-εξόδων. Δεν υπάρχει η δυνατότητα προσθήκης νέας ερώτησης στην πλατφόρμα, σε αντίθεση με τις υπόλοιπες κατηγορίες ερωτήσεων, στις οποίες πέρα από την επιλογή, έχει και την δυνατότητα της προσθήκης, αλλά και τροποποίησης κάποιας υπάρχουσας. Στις ερωτήσεις πολλαπλής επιλογής, η βαθμολόγηση επιτυγχάνεται σύμφωνα με τη σωστή απάντηση που έχει θέσει ο καθηγητής. Στις υπόλοιπες κατηγορίες δεν υπάρχει η δυνατότητα αυτόματης βαθμολόγησης. Στις ερωτήσεις ανάπτυξης ο εξεταστής γράφει την εκφώνηση, εφόσον δεν επιθυμεί να χρησιμοποιήσει κάποια από τις υπάρχουσες στην βιβλιοθήκη της πλατφόρμας, ενώ στις προσαρμοσμένες εργασίες, επισυνάπτει ένα αρχείο, το οποίο κατεβάζει ο εξεταζόμενος, και υποβάλλει την απάντησή του ανεβάζοντας την σε αρχείο στην πλατφόρμα ή στο GitHub.

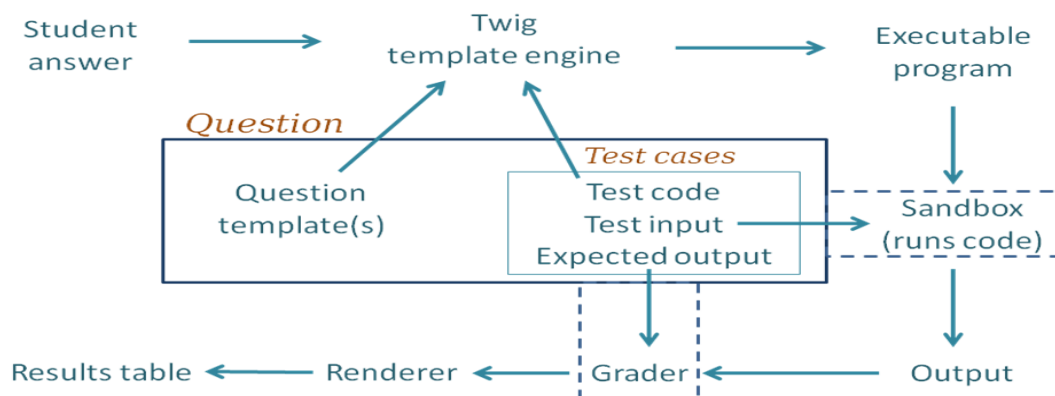
2.1.2. CodeRunner

Το CodeRunner είναι μία επέκταση για το Moodle που επιτρέπει στους καθηγητές να εκτελέσουν ένα πρόγραμμα για να βαθμολογήσουν την απάντηση ενός μαθητή, σε ερωτήσεις κώδικα. Χρησιμοποιείται σε μαθήματα προγραμματισμού όπου οι απαντήσεις των μαθητών βαθμολογούνται περνώντας από μία σειρά ελέγχων, που αφορούν κυρίως σετ εισόδων-εξόδων. Κατά κύριο λόγο, για να βαθμολογηθεί η εκάστοτε ερώτηση, θα πρέπει όλοι αυτοί οι έλεγχοι να έχουν περάσει επιτυχώς. Ωστόσο, είναι επίσης δυνατό να διαμορφωθούν οι ερωτήσεις του CodeRunner έτσι ώστε η βαθμολογία να καθορίζεται από τον αριθμό των ελέγχων που πέρασε επιτυχώς ο κώδικας.

Το CodeRunner υποστηρίζει τις γλώσσες προγραμματισμού Python, C, C++, Java, PHP, Pascal, JavaScript (NodeJS), Octave και MATLAB. Ωστόσο, υπάρχει η δυνατότητα να υποστηριχθούν επιπλέον γλώσσες χωρίς να τροποποιηθεί ο κώδικας του CodeRunner.

2.1.2.1 Διαδικασία Βαθμολόγησης

Η διαδικασία που ακολουθείται για την βαθμολόγηση (Richard Lobb, 2022) της εκάστοτε ερώτησης είναι η ακόλουθη και περιγράφεται στο παρακάτω διάγραμμα.



Εικόνα 1: Αρχιτεκτονική σχεδίαση Code Runner

Για κάθε απάντηση, ο κώδικας του μαθητή και οι περιπτώσεις ελέγχου που έχει θέσει ο καθηγητής μορφοποιούν το τελικό εκτελέσιμο πρόγραμμα. Αυτό επιτυγχάνεται με την χρήση του λογισμικού Twig που αφορά ένα λογισμικό το οποίο χρησιμοποιείται για την δημιουργία προτύπων ερώτησης. Έπειτα, το εκτελέσιμο πρόγραμμα μεταβιβάζεται στο Jobe Sandbox, το οποίο αποτελεί ένα εργαλείο που αναπτύχθηκε για την εκτέλεση κώδικα στις ερωτήσεις του CodeRunner και τρέχει σε εξωτερικό διακομιστή για λόγους ασφάλειας. Το πρόγραμμα μεταγλωττίζεται και εκτελείται στο Jobe Sandbox σύμφωνα με τις εισόδους που έχει θέσει ο εξεταστής. Η έξοδος του εκτελέσιμου προγράμματος μεταβιβάζεται στον βαθμολογητή, και συγκρίνεται με την έξοδο που έχει θέσει ο εξεταστής. Η διαδικασία αυτή επαναλαμβάνεται για κάθε περίπτωση ελέγχου της ερώτησης, και τα αποτελέσματα εμφανίζονται στον εξεταζόμενο σε πίνακα. Υπάρχει δυνατότητα τροποποίησης του βαθμολογητή από τον εξεταστή, υποβάλλοντας script κατά την διαμόρφωση της ερώτησης. Με αυτόν τον τρόπο πραγματοποιείται ένας πιο ενδελεχής έλεγχος, που μπορεί να αφορά επιπλέον περιπτώσεις ελέγχου, έλεγχο της μορφής που έχει η απάντηση, ή την ενδεχόμενη απαγόρευση ορισμένων μεθόδων για την απάντηση.

2.1.2.2 Τύποι ερωτήσεων

Το CodeRunner υποστηρίζει συγκεκριμένο αριθμό τύπων ερωτήσεων, ωστόσο μπορεί να επεκταθεί για την υποστήριξη περαιτέρω. Ο κάθε τύπος ερώτησης ορίζεται από το προφίλ της ερώτησης, στο οποίο ορίζονται οι παράμετροι που αφορούν την εκτέλεση του προγράμματος, όπως ο χρόνος εκτέλεσης, γλώσσα προγραμματισμού ή περιορισμοί μνήμης, καθώς και από την μορφή που θα έχει το εκτελέσιμο πρόγραμμα σύμφωνα με τις περιπτώσεις ελέγχου και τον κώδικα που έχει υποβληθεί από τον μαθητή. Ο τύπος ερώτησης καθορίζει επίσης και τον τρόπο με τον οποίο θα βαθμολογηθεί η απάντηση του εκάστοτε μαθητή. Το CodeRunner υποστηρίζει τέσσερις τύπους βαθμολογητών και ο εξεταστής καλείται να επιλέξει έναν από αυτούς. Η πιο απλή περίπτωση αφορά τον βαθμολογητή EquilityGrader, η χρήση του οποίου απαιτεί την πλήρη ταύτιση αναμενόμενης εξόδου που έχει θέσει ο εξεταστής, με την έξοδο της εκτέλεσης της απάντησης του μαθητή. Ο δεύτερος βαθμολογητής ονομάζεται NearEquilityGrader και αποτελεί μία απλή αναβάθμιση του πρώτου, αγνοώντας τα ενδεχόμενα κενά τα οποία μπορεί να περιέχει η εκάστοτε απάντηση. Ο βαθμολογητής RegexGrader διαβάζει την αναμενόμενη έξοδο του προγράμματος ως regular expression, και προσπαθεί να την ταιριάξει με οποιοδήποτε σημείο της εξόδου του προγράμματος εκτέλεσης. Τέλος, η πιο πολύπλοκη, αλλά και ευέλικτη μορφή βαθμολογητή που υποστηρίζει το CodeRunner αποτελεί ο Template Grader. Η χρήση αυτού του βαθμολογητή απαιτεί από τον εξεταστή να υποβάλει κώδικα συγκεκριμένης μορφής. Τα πλεονεκτήματα που προσφέρει είναι η ευελιξία στην βαθμολογία της ερώτησης, παρέχοντας την βαθμολόγηση της κάθε περίπτωσης ελέγχου ξεχωριστά, καθώς και η μορφοποίηση του πίνακα αποτελεσμάτων που επιστρέφουν στον εξεταζόμενο μετά την υποβολή της κάθε ερώτησης.

2.1.2.3 CodeRunner Sandboxing- Jobe Server

Ο Jobe Server (Lobb, 2022) είναι ένας διακομιστής που υποστηρίζει την μεταγλώττιση και την εκτέλεση κώδικα σε διάφορες γλώσσες προγραμματισμού. Αναπτύχθηκε ως ένα sandbox για το CodeRunner, αλλά θα μπορούσε να εφαρμοστεί και σε άλλα πλαίσια στον τομέα της εκπαίδευσης. Για την εκτέλεση κώδικα στον Jobe, καθορίζονται η γλώσσα προγραμματισμού, ο κώδικας που θα εκτελεστεί ή και θα μεταγλωττιστεί, η είσοδος του προγράμματος εκτέλεσης και προαιρετικά μία λίστα με επιπλέον αρχεία. Μετά την εκτέλεση, επιστρέφονται οι πληροφορίες της εκτέλεσης μαζί με την έξοδο του προγράμματος και τα ενδεχόμενα σφάλματα που προκύπτουν από αυτή. Η επικοινωνία μεταξύ του Jobe και του CodeRunner επιτυγχάνεται με την χρήση Restful API, χρησιμοποιώντας τα ακόλουθα endpoints (Lobb, Jobe: REST API, 2015):

Request Name	HTTP Type	Target Resource	Parameters (* denotes required)	Possible response codes	Comment
submit_run	POST	/runs	run_spec*	200 OK 202 Accepted 400 BadRequest 404 Not Found	A return code of 200 is accompanied by the run result; 202 denotes the job has been queued for later execution (notes 1, 2, 6).
get_run_status	GET	/runresults/id		200 OK 204 NoContent 400 BadRequest 404 NotFound	A return code of 200 is accompanied by the run result; 204 denotes the job is still pending (notes 2, 6).
get_languages	GET	/languages		200 OK 400 BadRequest	Returns a JSON encoded list of supported languages (note 3).
put_file	PUT	/files/uniqueid	file_contents*	204 No content 400 BadRequest 403 Forbidden	It is the client's responsibility to ensure a unique file ID (notes 4, 6, 7, 8)

Εικόνα 2: Requests του Jobe Server

Για την λειτουργία του Sandbox ο Jobe χρησιμοποιεί το σύστημα DOMjudge (Jaap Eldering, 2004) που αποτελεί ένα αυτόματο σύστημα για την εκτέλεση κώδικα με κύρια εφαρμογή σε διαγωνισμούς προγραμματισμού. Πρόκειται για ένα open source λογισμικό με δυνατότητα επέκτασης ή τροποποίησης με την χρήση Rest API. Η χρήση του DOMjudge απαιτεί λογισμικό Linux με δικαιώματα διαχειριστή, διακομιστή Apache Server με Php γραμμή εντολών, MySQL ή MariaDB βάση δεδομένων καθώς και compilers για τις γλώσσες προγραμματισμού που θα χρησιμοποιηθούν. Η εκτέλεση του κώδικα γίνεται με κατανεμημένο τρόπο με την χρήση εικόνων Docker. Τέλος, επιβάλλονται περιορισμοί που αφορούν κυρίως τους πόρους του συστήματος, όπως περιορισμούς στην μνήμη, στον αριθμό διαδικασιών και στον χρόνο εκτέλεσης του προγράμματος, χωρίς ωστόσο να επιβάλλονται περιορισμοί στις εντολές συστήματος.

2.1 Συστήματα Διαχείρισης Μάθησης

2.2.1 Moodle

Η πλατφόρμα Moodle είναι ένα Εικονικό Περιβάλλον Μάθησης Ανοιχτού Κώδικα. Υποστηρίζει την διεξαγωγή ηλεκτρονικών μαθημάτων μέσω διαδικτύου και προσφέρει υπηρεσίες ασύγχρονης τηλεκπαίδευσης. Διαθέτει πληθώρα εργαλείων για την υποστήριξη της διαδικτυακής διδασκαλίας βοηθώντας τους εκπαιδευτές να διαχειρίζονται τους πόρους τους, δηλαδή αρχεία ή μαθήματα και να υποβάλλουν εργασίες και διαγωνίσματα μαζί με τις αντίστοιχες βαθμολογίες τους για κάθε μάθημα. Παράλληλα, τα εργαλεία οι εκπαιδευόμενοι έχουν πρόσβαση σε αυτό το διαδικτυακό υλικό. Η πλατφόρμα Moodle ενθαρρύνει την ομαδική εργασία παρέχοντας ομάδες συζητήσεων σε forums ή ιστολόγια.

Οι χρήστες της πλατφόρμας Moodle διακρίνονται σε Διαχειριστές, Εκπαιδευτικούς, Εκπαιδευόμενοι και Επισκέπτες. Οι διαχειριστές έχουν την πλήρη εποπτεία της εφαρμογής και διαχειρίζονται όλες τις παραμέτρους της πλατφόρμας. Οι εκπαιδευτές είναι οι διδάσκοντες που δημιουργούν τα μαθήματα και προσθέτουν το εκπαιδευτικό υλικό στην πλατφόρμα. Επίσης μπορεί να επικοινωνεί με τους εκπαιδευόμενους δημιουργώντας ομάδες εργασίας. Με την χρήση της πλατφόρμας δημιουργεί και επεξεργάζεται μαθήματα, υποβάλλει βαθμολογία στους εκπαιδευόμενους και επεξεργάζεται τις δραστηριότητές τους. Οι εκπαιδευόμενοι είναι οι χρήστες που είναι εγγεγραμμένοι σε τουλάχιστον ένα μάθημα και έχουν πρόσβαση στο υλικό και τις δραστηριότητες του μαθήματος που είναι εγγεγραμμένοι. Τέλος, οι επισκέπτες είναι οι χρήστες που δεν έχουν λογαριασμό στην πλατφόρμα, και έχουν την δυνατότητα να πλοηγηθούν στο εκπαιδευτικό υλικό της πλατφόρμας, χωρίς να έχουν την δυνατότητα συμμετοχής στις εκπαιδευτικές δραστηριότητες των μαθημάτων.

Συμπερασματικά, το σύστημα Moodle αποτελεί μια πλατφόρμα με πλούσιο υλικό στον χώρο της εξ' αποστάσεως εκπαίδευσης. Σε σχέση με αντίστοιχα συστήματα περιέχει περισσότερες επιλογές και δυνατότητες και πολύ μεγάλη ευελιξία. Ωστόσο για την πλήρη αξιοποίηση των πόρων και των λειτουργιών του απαιτείται εκπαίδευση από τους χρήστες του (ElearningLessonsGreece, -).

2.2.2 Open eClass

Η πλατφόρμα Open eClass είναι ένα ολοκληρωμένο Σύστημα διαχείρισης ηλεκτρονικών μαθημάτων για την οργάνωση, αποθήκευση και παρουσίαση εκπαιδευτικού υλικού. Στόχος της πλατφόρμας αποτελεί η ενσωμάτωση των νέων τεχνολογιών και η εποικοδομητική χρήση του διαδικτύου στην εκπαίδευση. Αποτελεί ένα σύστημα ανοιχτού κώδικα και διανέμεται ελεύθερα. Το Open eClass υποστηρίζει τις υπηρεσίες ασύγχρονης τηλεκπαίδευσης χωρίς περιορισμούς. Τα βασικά χαρακτηριστικά της πλατφόρμας είναι οι διακριτοί ρόλοι χρηστών, οι διακριτές κατηγορίες μαθημάτων, η σταθερότητα και η αξιοπιστία.

Το σύστημα υποστηρίζει τρεις κύριες κατηγορίες χρηστών, τους εκπαιδευτές, τους εκπαιδευόμενους και τους διαχειριστές. Ο εκπαιδευτής είναι υπεύθυνος για τη δημιουργία και τη διαχείριση των ηλεκτρονικών μαθημάτων. Μπορεί να δημιουργήσει όσα μαθήματα επιθυμεί και εισάγει εκπαιδευτικό υλικό για το κάθε μάθημα και έχει την δυνατότητα να επικοινωνεί με τους εκπαιδευόμενους

δημιουργώντας ομάδες εργασίας, ομαδικές συζητήσεις και γενικά ελέγχοντας την εκπαιδευτική διαδικασία. Ο εκπαιδευόμενος μπορεί να εγγραφεί σε μαθήματα και να έχει πρόσβαση στο εκπαιδευτικό υλικό τους, στις ομάδες εργασίας και στις περιοχές συζητήσεων. Τέλος, οι διαχειριστές είναι υπεύθυνοι για την επίβλεψη της λειτουργίας της πλατφόρμας. Έχουν την δυνατότητα να δημιουργούν και να ελέγχουν τους λογαριασμούς των χρηστών της πλατφόρμας, ενώ παράλληλα διαχειρίζονται την βάση δεδομένων και τον εξυπηρετητή. Αξίζει να σημειωθεί πως η πλατφόρμα Open eClass υποστηρίζει και την ύπαρξη δευτερευόντων ρόλων όπως βοηθός διαχειριστή, βοηθός καθηγητή, υπεύθυνος ομάδας, επισκέπτης κ.α. Το Open eClass αποτελεί μια πολύ απλή πλατφόρμα για την στοιχειώδη διαχείριση ψηφιακού εκπαιδευτικού περιεχομένου. Το βασικό της πλεονέκτημα είναι η ευκολία και η άνεση στην χρήση, ωστόσο όταν ο χρήστης εντάξει τις βασικές λειτουργίες e-learning και χρειαστεί να τις εξελίξει, θα πρέπει να χρησιμοποιήσει άλλη πλατφόρμα.

3. Χρησιμοποιούμενα Εργαλεία και Τεχνολογίες

3.1 Frameworks

3.1.1 Spring-Boot

Το Spring-Boot είναι ένα framework ανοιχτού κώδικα βασισμένο στην γλώσσα προγραμματισμού Java και χρησιμοποιείται για την ανάπτυξη μικροϋπηρεσιών. Παρέχει μια εξαιρετική πλατφόρμα για Java προγραμματιστές, ώστε να δημιουργήσουν μια αυτόνομη εφαρμογή Spring έτοιμη για παραγωγή. Οι χρήστες ξεκινούν με ελάχιστη διαμόρφωση χωρίς να απαιτείται η πλήρης και πολύπλοκη διαμόρφωση που απαιτεί η χρήση του Spring πλαισίου.

3.1.1.1 Πλεονεκτήματα και στόχοι

Η χρήση του Spring Boot πλαισίου παρέχει έναν κατανοητό και εύκολο τρόπο δημιουργίας εφαρμογών. Έτσι αυξάνεται η παραγωγικότητα και μειώνεται ο χρόνος που απαιτείται για την ανάπτυξη. Σχεδιάστηκε με σκοπό να μη χρειάζονται πολλές ρυθμίσεις XML όπως στο Spring framework και να διευκολύνεται η δημιουργία Spring εφαρμογών έτοιμες για παραγωγή. Επίσης παρέχει έναν ευέλικτο τρόπο ρύθμισης Java Beans, Xml διαμόρφωσης και επικοινωνίας με Βάση Δεδομένων. Τέλος υποστηρίζει επικοινωνία μέσω Rest API και παρέχει ενσωματωμένο Servlet Container. (TutorialsPoint)

3.1.1.2. Servlet Container

Το Servlet είναι στοιχείο ανάπτυξης του πλαισίου JEE (Java Enterprise Edition) και χρησιμοποιείται στην ανάπτυξη διαδικτυακών εφαρμογών. Ουσιαστικά πρόκειται για πρόγραμμα σε Java που εκτελείται μέσα σε Container. Τα Servlets παρέχουν λειτουργικότητα διακομιστή και είναι υπεύθυνα για την αποδοχή αιτημάτων και την επεξεργασία και την ανταπόκρισή τους. Το Servlet είναι έτοιμο να δεχτεί αιτήματα μόλις ολοκληρωθεί η αρχικοποίηση της Spring Boot εφαρμογής. (Baeldung, Introduction to Servlets and Servlet Containers)

3.1.1.3. Java Beans

Τα Java Beans αποτελούν σημαντικά στοιχεία του Spring Framework. Είναι αντικείμενα που αποτελούν την ραχοκοκαλιά της εκάστοτε εφαρμογής και δημιουργούνται και συντηρούνται από το Spring IoC (inversion of control) Container. Αν και δεν υπάρχει σαφής και συγκεκριμένος ορισμός για τα Java Beans, ουσιαστικά αποτελούν κλάσεις που ενσωματώνουν πολλά αντικείμενα σε ένα μόνο αντικείμενο, το bean, και στοχεύουν στη δημιουργία και την επαναχρησιμοποίηση στοιχείων για την εφαρμογή. Η εισαγωγή και η ενσωμάτωση αντικειμένων σε Spring Container ονομάζεται Inversion of Control και είναι η διαδικασία κατά την οποία το κάθε αντικείμενο ορίζει τις εξαρτήσεις του, χωρίς να τις υλοποιεί. (Baeldung, What is a Spring Bean?)

3.1.2. ASP.NET

Το Asp.Net είναι ένα framework της Microsoft που χρησιμοποιείται για την ανάπτυξη web εφαρμογών. Η κύρια γλώσσα προγραμματισμού που υποστηρίζεται είναι η C#. Το framework της Asp.Net παρέχει πληθώρα εργαλείων που βοηθούν τους προγραμματιστές στην γρήγορη και εύκολη ανάπτυξη δυναμικών ιστοσελίδων, οι οποίες τρέχουν server side στον IIS σε Microsoft Windows.

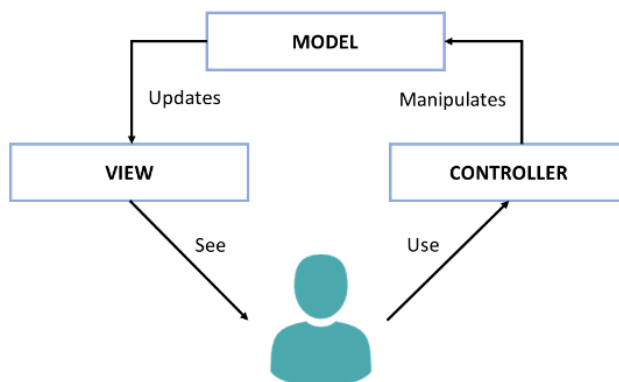
3.1.2.1 ASP.NET Core MVC

Το αρχιτεκτονικό μοτίβο Model-View-Controller ή αλλιώς MVC, είναι ένα σχεδιαστικό μοτίβο ανάπτυξης εφαρμογών, που συναντάται σε διάφορες γλώσσες προγραμματισμού, όπως Java, C και C#. Χωρίζει την εφαρμογή σε τρεις κύριες κατηγορίες στοιχείων, τα μοντέλα (Model), τις σελίδες (Views) και τους ελεγκτές (Controllers), και χρησιμοποιείται για την δημιουργία περιβαλλόντων αλληλεπίδρασης στον χρήστη (Smith, 2022) (τj45, 2008). Χρησιμοποιώντας αυτό το μοτίβο, τα αιτήματα των χρηστών δρομολογούνται στον controller που είναι υπεύθυνος με την αλληλεπίδραση με το μοντέλο, για την εκτέλεση ενεργειών του χρήστη και την ανάκτηση δεδομένων από την βάση. Ο controller επίσης είναι υπεύθυνος για την επιλογή της σελίδας που θα εμφανίζεται στο γραφικό περιβάλλον του χρήστη και παρέχει όλα τα δεδομένα και τις τιμές των μοντέλων που απαιτούνται.

3.1.2.1.1. Αλληλεπίδραση κατηγοριών MVC

Η συνεργασία των τριών κατηγοριών στοιχείων του MVC μοτίβου επιτυγχάνεται ως εξής. Αρχικά, ο controller είναι το στοιχείο που χειρίζεται την αλληλεπίδραση της εφαρμογής με τον χρήστη. Τροποποιεί το μοντέλο ανάλογα με τις ενέργειες του χρήστη και ενημερώνει την κατάστασή του. Είναι το αρχικό σημείο εκκίνησης, και είναι υπεύθυνος για την επιλογή των μοντέλων που θα συνεργαστεί και των σελίδων που θα εμφανίσει στο χρήστη. Οι σελίδες είναι υπεύθυνες για την παρουσίαση του περιεχομένου της εφαρμογής στο γραφικό περιβάλλον του χρήστη. Περιέχουν ελάχιστη λογική που αφορά μόνο στην παρουσίαση μοντέλων στον χρήστη. Οι .NET εφαρμογές χρησιμοποιούν τη μηχανή προβολής Razor και ενσωματώνουν c# κώδικα στην HTML σήμανση. Καταλήγοντας, τα μοντέλα στις MVC εφαρμογές

αντιπροσωπεύουν την κατάσταση της εφαρμογής και τις όποιες λειτουργίες που πρέπει να εκτελεστούν σε αυτή και ενημερώνουν τις αντίστοιχες σελίδες και controllers όταν υπάρχει αλλαγή στα δεδομένα, επιτρέποντας παράλληλα στις σελίδες να ενημερώνουν τη γραφική απεικόνιση στο χρήστη. (Razi)



Εικόνα 3: Αλληλεπίδραση Model-View-Controller

3.1.2.1.2. Routing

Πρόκειται για ένα ισχυρό εργαλείο του ASP.NET Core MVC και αφορά στην αντιστοίχιση του URL με τον αντίστοιχο controller, επιτρέποντας την δημιουργία εφαρμογών με κατανοητά και εύκολα στην αναζήτηση URL. Βοηθά στην βελτιστοποίηση των μηχανών αναζήτησης, χωρίς να λαμβάνεται υπόψη ο τρόπος οργάνωσης των αρχείων στον διακομιστή.

3.1.2.1.3. Areas

Οι περιοχές areas παρέχουν έναν τρόπο κατανομής μιας μεγάλης web εφαρμογής σε μικρότερες λειτουργικές ομαδοποιήσεις. Κάθε περιοχή αποτελεί και μια δομή MVC μέσα στην εφαρμογή. Στις εφαρμογές που ακολουθούν το μοτίβο MVC, τα στοιχεία model, view και controller διατηρούνται σε διαφορετικούς φακέλους, και διατηρούνται συμβάσεις στην ονομασία για την αλληλεπίδραση των στοιχείων αυτών.

3.1.2.1.4. Filters

Τα φίλτρα χρησιμοποιούνται από τους προγραμματιστές κυρίως για τη διαχείριση εξαιρέσεων και την εξουσιοδότηση. Είναι μια προσαρμοσμένη κλάση στην οποία γράφεται λογική που εκτελείται πριν ή μετά την εκτέλεση μιας μεθόδου. Εφαρμόζονται σε μία μέθοδο ή σε έναν controller είτε δηλώνοντας ένα χαρακτηριστικό ενός φίλτρου, είτε εφαρμόζοντας μια διεπαφή του.

3.1.2.2. Identity Framework

Το Identity Framework είναι ένα framework της ASP.NET που χρησιμοποιείται για την ταυτοποίηση και την εξουσιοδότηση χρηστών σε ένα σύστημα. Η δομή αυτή επιτρέπει την προσθήκη προσαρμοσμένης λειτουργικότητας όσον αφορά την λειτουργία σύνδεσης/εγγραφής καθώς και στην προσαρμογή δεδομένων σχετικά με τον εκάστοτε συνδεδεμένο χρήστη. Αποθηκεύει τις πληροφορίες των χρηστών σε βάση, αλλά είναι δυνατή και η χρήση διαφόρων παροχών αποθήκευσης, όπως οι υπηρεσίες SharePoint και Azure table services. Η εξουσιοδότηση βάσει ρόλου, ή role base authorization, αποτελεί σημαντική λειτουργία για το Identity, καθώς μπορούμε να δημιουργήσουμε και να διαχωρίσουμε διάφορους ρόλους χρηστών με διαφορετικά δικαιώματα. Έτσι μας επιτρέπει να προσθέτουμε χρήστες σε κάποιον ρόλο και μπορούμε να τους περιορίζουμε σε συγκεκριμένα μέρη της εφαρμογής. Χρησιμοποιώντας το Identity Framework κατά την δημιουργία μιας εφαρμογής, εισάγονται πακέτα που είναι υπεύθυνα για την εισαγωγή των δεδομένων του Identity και της δομής της βάσης σε SQL server, για την παροχή βασικών διεπαφών που διαχειρίζονται την λειτουργικότητα των λειτουργιών σύνδεσης και εγγραφής καθώς και για τη λειτουργικότητα για τον έλεγχο ταυτότητας.

3.1.2.2.1 Βασικοί πυλώνες του Identity Framework

Ο χρήστης του συστήματος αντιπροσωπεύεται από ένα αντικείμενο “user” όπου περιέχεται ο βασικός έλεγχος ταυτότητας με ένα συνθηματικό (password) και μία λέξη (username). Ο έλεγχος αυτός καταγράφεται από την κλάση Identity User, η οποία μπορεί να κληρονομηθεί από προσαρμοσμένη κλάση που χρησιμοποιείται για την καταγραφή πληροφοριών προφίλ. Για την διαχείριση των χρηστών χρησιμοποιείται η κλάση UserManager. Στην κλάση αυτή περιέχονται αρκετές λειτουργίες, όπως δημιουργία ή διαγραφή χρηστών, αλλαγή ονόματος και κωδικού καθώς και η εισαγωγή ή αφαίρεση ρόλων από συγκεκριμένους χρήστες.

Σε μια εφαρμογή μπορεί να υπάρχουν κάποιοι χρήστες με διαφορετικούς ρόλους από κάποιους άλλους. Το αντικείμενο “role” αντιπροσωπεύει τον ρόλο που έχει ο εκάστοτε χρήστης στο σύστημα. Για την διαχείριση των διαφορετικών ρόλων των χρηστών χρησιμοποιείται η κλάση role manager και είναι υπεύθυνη για την δημιουργία ή την αφαίρεση ενός ρόλου καθώς και για τον έλεγχο της ύπαρξης των ρόλων στο σύστημα.

Σημειώνεται πως οι παραπάνω κλάσεις διαχειρίζονται τους χρήστες και τους ρόλους τους, και δεν υλοποιούν την ταυτοποίηση των χρηστών. Για αυτόν τον λόγο χρησιμοποιείται η κλάση Authentication Manager, η οποία παρέχει τις λειτουργίες ταυτοποίησης, όπως εγγραφή, σύνδεση και αποσύνδεση.

3.1.2.2. Entity Framework

Πρόκειται για ένα framework που παρέχει η Microsoft και αφορά την αυτοματοποίηση όλων των ενεργειών που σχετίζονται με την βάση δεδομένων μιας εφαρμογής. Είναι το πιο διαδεδομένο ORM (object relational mapper) για τη γλώσσα προγραμματισμού C#, δηλαδή αποτελεί εργαλείο αντιστοίχισης αντικειμένων σε πίνακες σε βάση δεδομένων στις αντικειμενοστραφείς γλώσσες προγραμματισμού. Επιτρέπει στους προγραμματιστές να διαχειρίζονται δεδομένα χρησιμοποιώντας αντικείμενα συγκεκριμένων κλάσεων. Δεν εστιάζουν στους πίνακες της βάσης δεδομένων, αλλά διαχειρίζονται αυτά τα δεδομένα σε ένα υψηλότερο επίπεδο και έτσι απαιτείται λιγότερος κώδικας και πολυπλοκότητα σε σύγκριση με τον παραδοσιακό τρόπο ανάπτυξης εφαρμογών.

Το Entity Framework συνδέει τα αντικείμενα των κλάσεων της εφαρμογής με τη βάση δεδομένων. Αποθηκεύει τα δεδομένα και τις ιδιότητες των αντικειμένων στη βάση, καθώς και ανακτά δεδομένα μετατρέποντάς τα αυτόματα σε αντικείμενα. Η πρόσβαση στα δεδομένα πραγματοποιείται χρησιμοποιώντας ένα μοντέλο. Το μοντέλο αποτελείται από κλάσεις οντοτήτων, και ένα αντικείμενο context που αντιπροσωπεύει μια συνεδρία με τη βάση δεδομένων. Το αντικείμενο αυτό επιτρέπει την αναζήτηση και την αποθήκευση των δεδομένων στη βάση.

3.1.2.2.1. Χαρακτηριστικά του Entity Framework

Το Entity Framework είναι ένα cross-platform framework που τρέχει σε Windows, Linux και Mac. Έχει ως βασικό χαρακτηριστικό την χρήση LINQ Queries για την ανάκτηση δεδομένων από τη βάση. Τα LINQ Queries αποτελούν εκφράσεις σε C#, οι οποίες μεταφράζονται στην κατάλληλη γλώσσα, ανάλογα με την βάση δεδομένων που χρησιμοποιείται. Επιπλέον παρέχει ένα σύνολο Migration εντολών που εκτελούνται στην κονσόλα Nugget Package Manager του Visual Studio ή στην γραμμή εντολών. Οι εντολές αυτές χρησιμοποιούνται για την διαχείριση ή την δημιουργία του σχήματος της βάσης Δεδομένων.

Η υποβολή αιτημάτων και η αποθήκευση δεδομένων επιτυγχάνονται με την χρήση μοντέλων με get-set ιδιότητες από διαφορετικούς τύπους δεδομένων. Η εισαγωγή, η διαγραφή και ενημέρωση στοιχείων στη βάση επιτυγχάνονται μέσα από εντολές του Entity Framework και τα δεδομένα αποθηκεύονται προσωρινά, ώστε να μην απαιτείται πρόσβαση στη βάση δεδομένων όταν εκτελούνται επαναλαμβανόμενα ερωτήματα. Τέλος, το Entity Framework καταγράφει τις αλλαγές που εμφανίζονται στα στιγμιότυπα των οντοτήτων που υποβάλλονται στη βάση και προστατεύει από την επεγράφη δεδομένων από διαφορετικούς χρήστες χρησιμοποιώντας την λειτουργία Optimistic Concurrency από προεπιλογή. (Tutorial)

3.2. Επικοινωνία εφαρμογών μέσω REST αρχιτεκτονικής

Η τεχνολογία REST αποτελεί ένα σύνολο αρχιτεκτονικών περιορισμών που χρησιμοποιείται με διάφορους τρόπους από προγραμματιστές API. Εφευρέθηκε από τον Roy Fielding και τα αρχικά REST σημαίνουν μεταφορά κατάστασης αναπαράστασης. Όταν πραγματοποιείται αίτημα από έναν Client μέσω REST API, μεταφέρεται ένα σύνολο πληροφοριών στον αιτούντα. Αυτές οι πληροφορίες παραδίδονται σε διάφορες μορφές μέσω του πρωτοκόλλου HTTP. Η πιο δημοφιλής μορφή αρχείου που χρησιμοποιείται για τη μεταφορά πληροφοριών είναι η JSON (Sletton, 2020).

3.2.1. Διεπαφή API

Μια διεπαφή προγραμματισμού εφαρμογών είναι μια συλλογή πρωτοκόλλων για την επικοινωνία και την ενοποίηση εφαρμογών λογισμικού. Περιγράφεται ως σύμβαση μεταξύ ενός συστήματος που λειτουργεί ως sever με ένα σύστημα που έχει λειτουργία client. Το σύστημα client, υποβάλλει αίτημα στον server ζητώντας το περιεχόμενο που πρέπει να του προσφέρει, και ο server επιστρέφει το περιεχόμενο αυτό ως απάντηση.

3.2.2. RESTful API

Το RESTful API είναι τύπος διεπαφής API που συμμορφώνεται με τους κανόνες και τους περιορισμούς του αρχιτεκτονικού μοτίβου Rest, και επιτρέπει την αλληλεπίδραση με Restfull εφαρμογές διαδικτύου. Ένα RESTful API χρησιμοποιεί αιτήματα για τη λήψη, τη δημοσίευση, την επεξεργασία και τη διαγραφή δεδομένων. Τέλος τα συστήματα που αλληλοεπιδρούν μέσω REST API μπορεί να έχουν πολλές διαφορές, όπως γλώσσα προγραμματισμού, λειτουργικό σύστημα ή βάση δεδομένων.

3.3. Docker

Το Docker είναι μια πλατφόρμα λογισμικού ανοιχτού κώδικα που χρησιμοποιείται για την ανάπτυξη και την εκτέλεση εφαρμογών σε επίπεδο Λειτουργικού Συστήματος. Προσφέρει αυτοματοποιημένες διαδικασίες για την ανάπτυξη εφαρμογών, σε απομακρυσμένες από το χρήστη περιοχές που ονομάζονται Software Containers. Βασίζεται στο λογισμικό Linux και επιτρέπει ανεξάρτητα Software Containers να εκτελούνται παράλληλα στο ίδιο λειτουργικό σύστημα. Αυτό έχει ως αποτέλεσμα να αποφεύγεται η χρήση επιπλέον υπολογιστικών πόρων και να μειώνεται αισθητά η καθυστέρηση μεταξύ της σύνταξης του κώδικα μιας εφαρμογής και της εκτέλεσής της στην παραγωγή.

3.3.1. Πώς λειτουργεί το λογισμικό Docker

Η πλατφόρμα Docker επιτρέπει την αποθήκευση και την εκτέλεση μιας εφαρμογής σε περιβάλλον απομονωμένο από τον χρήστη. Το περιβάλλον αυτό ονομάζεται Container. Η απομόνωση αυτή δίνει την δυνατότητα της παράλληλης εκτέλεσης πολλών Container από τον ίδιο υπολογιστή. Τα container περιέχουν όλους τους πόρους που απαιτούνται για την εκτέλεση και την αποθήκευση της εκάστοτε εφαρμογής, χωρίς να επιβαρύνουν σημαντικά το κεντρικό σύστημα. Σημειώνεται πως ο χρήστης μπορεί εύκολα να μοιράζεται τις εφαρμογές που έχει αναπτύξει ως Containers, με άλλους χρήστες οι οποίοι θα λαμβάνουν το ίδιο Container και η εφαρμογή θα λειτουργεί με τον ίδιο ακριβώς τρόπο και στο σύστημά τους. (Docs)

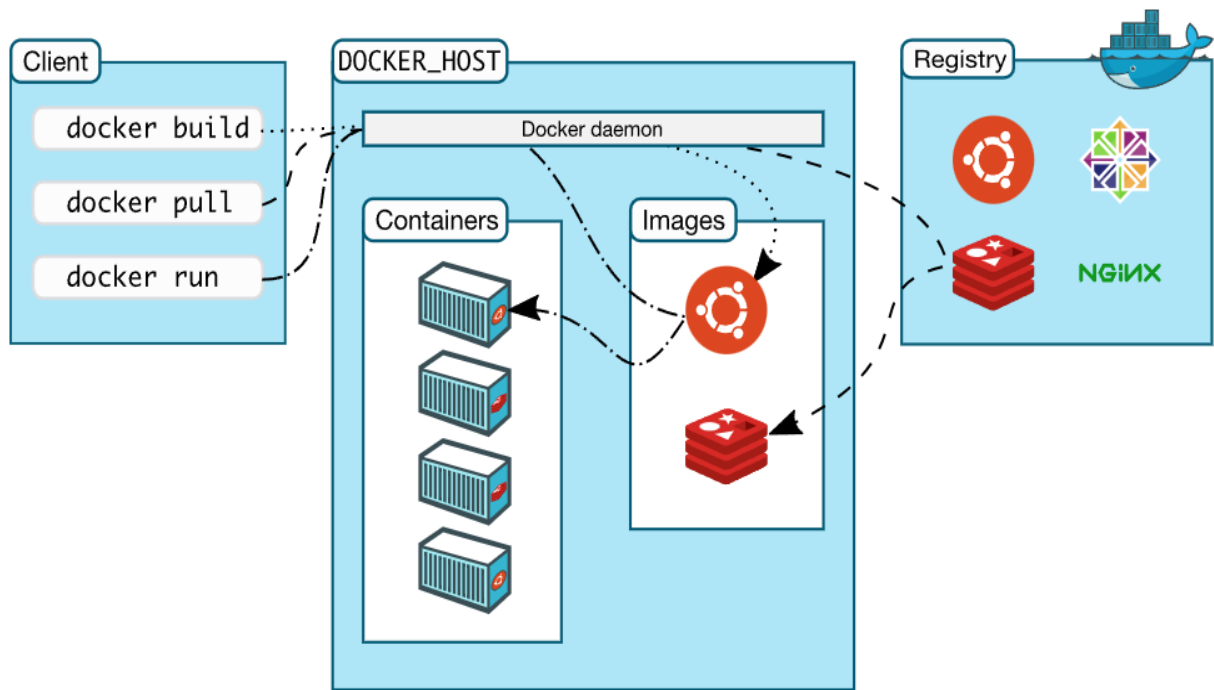
3.3.2. Τα δομικά στοιχεία του Docker

Για τη χρήση του εργαλείου Docker είναι απαραίτητη η χρήση ενός Dockerfile, μιας εικόνας (Image) και τουλάχιστον ενός Container. Η εικόνα περιέχει τις οδηγίες για τη δημιουργία ενός Container. Συχνά μια εικόνα βασίζεται σε μια άλλη εικόνα. Ο χρήστης μπορεί να δημιουργήσει μια εικόνα που βασίζεται σε μια υπάρχουσα εγκαθιστώντας σε αυτή τις λεπτομέρειες διαμόρφωσης της εφαρμογής του. Για την δημιουργία μιας εικόνας απαιτείται η σύνταξη ενός αρχείου, που ονομάζεται dockerfile. Σε αυτό το αρχείο καθορίζονται τα βήματα που απαιτούνται για την δημιουργία της εικόνας και την εκτέλεσή της με την μορφή εντολών. Κάθε εντολή που περιέχει το dockerfile, δημιουργεί ένα νέο layer στην εικόνα. Αξίζει να σημειωθεί πως κάθε αλλαγή στο dockerfile αναδημιουργεί μόνο τα επίπεδα που έχουν αλλάξει, καθιστώντας τις εικόνες πολύ ελαφριές και γρήγορες σε σχέση με άλλες τεχνολογίες εικονοποίησης.

Για να εκτελεστεί μια εικόνα χρειάζεται ένα στιγμιότυπο με δυνατότητες εκτέλεσης. Το στιγμιότυπο αυτό ονομάζεται Container. Το container ορίζεται από την εικόνα, καθώς και από τες ενδεχόμενες επιλογές διαμόρφωσης που παρέχει ο χρήστης κατά την δημιουργία ή την εκκίνησή του. Όταν διαγραφεί ένα container, όλες οι αλλαγές στην κατάστασή του που δεν αποθηκεύονται, εξαφανίζονται. Ο χρήστης μπορεί να δημιουργήσει, να ξεκινήσει, να σταματήσει και να διαγράψει ένα Container χρησιμοποιώντας το Docker API. Αν, κατά την εκτέλεση ενός container υπάρχει η ανάγκη για τη διατήρηση και την αποθήκευση δεδομένων, τότε είναι απαραίτητη η χρήση των docker volumes. Τα δεδομένα αποθηκεύονται στον κεντρικό υπολογιστή, συνήθως στο filesystem, και η διαχείριση των volumes γίνεται αποκλειστικά από το Docker.

3.3.3. Η Αρχιτεκτονική του Docker

Το Docker χρησιμοποιεί την server-client αρχιτεκτονική. Ο Docker client επικοινωνεί με τον Docker Deamon, ο οποίος αναλαμβάνει την κατασκευή, την λειτουργία και την κατανομή των Docker Containers. Η επικοινωνία μεταξύ τους επιτυγχάνεται χρησιμοποιώντας REST API. Η τοποθεσία όπου αποθηκεύονται οι εικόνες Docker ονομάζεται Docker Registry. Από προεπιλογή, οι εικόνες αποθηκεύονται δημόσια στο Docker Hub, αλλά υπάρχει η δυνατότητα να αποθηκευτούν σε ιδιωτικό αρχείο.



Εικόνα 4: Αρχιτεκτονική σχεδίαση Docker

3.4. Learning Tools Interoperability

Η τεχνολογία Learning Tool Interoperability αναπτύχθηκε από την IMS Global Learning Consortium και καθορίζει έναν εύκολο και ασφαλή τρόπο σύνδεσης συστημάτων εκμάθησης, ή learning management system (LMS), με εξωτερικές εφαρμογές. Ο εκπαιδευόμενος δεν συνδέεται με το εξωτερικό σύστημα, αλλά το εκάστοτε LMS μοιράζεται τις πληροφορίες του μαζί με το πλαίσιο μάθησης με αυτό. Το 2019 ανακοινώθηκε η διαθεσιμότητα της έκδοσης LTI 1.3 και των υπηρεσιών που αποτελούν το LTI Advantage. Αυτή η έκδοση είναι μια ενημέρωση του LTI 1.1, και καταργεί όλες τις προηγούμενες εκδόσεις LTI, συμπεριλαμβανομένων των LTI 1.1, LTI 1.2 και LTI 2.0.

Το LTI Advantage προσφέρει περισσότερη ασφάλεια, χρησιμοποιώντας το πρωτόκολλο OAuth 2.0 και Json Web Token, καθώς και έναν σαφέστερο τρόπο σύνδεσης των πλατφόρμων εκμάθησης και των εξωτερικών εργαλείων. Οι υπηρεσίες που προστίθενται στο πλαίσιο ασφάλειας του LTI 1.3 είναι οι ακόλουθες:

- Names and Role Provisioning Services
- Assignment and Grade Services
- Deep Linking

Οι υπηρεσίες αυτές αντικαθιστούν ορισμένες υπηρεσίες που προσφέρονται από τις προηγούμενες εκδόσεις LTI και αυξάνουν την διαλειτουργικότητα μεταξύ των LMS και των εξωτερικών εργαλείων εκμάθησης.

3.4.1. Names and Role Provisioning Services

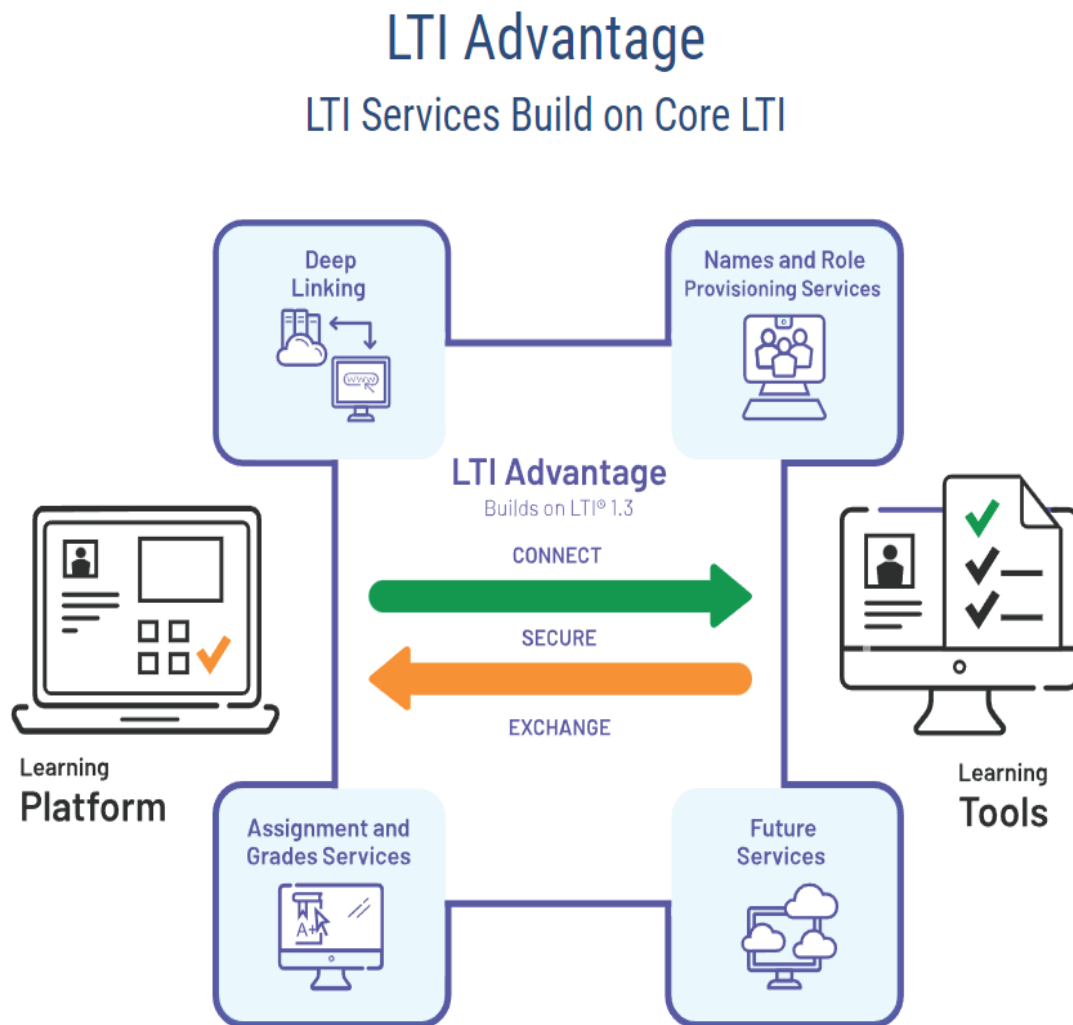
Η υπηρεσία παροχής ονομάτων και ρόλων είναι μια υπηρεσία του LTI Advantage που παρέχει στο εργαλείο πρόσβαση στην λίστα των χρηστών και τους ρόλους τους σε ένα συγκεκριμένο πλαίσιο, όπως για παράδειγμα μια εξέταση μαθήματος. Πρόκειται για μια αναβάθμιση της αντίστοιχης υπηρεσίας της έκδοσης Lti 1.1, στην οποία οι καθηγητές δεν είχαν πρόσβαση στην πλήρη λίστα των μαθητών που ήταν εγγεγραμμένοι για κάποια δραστηριότητα, και έτσι δεν ήταν δυνατόν να γνωρίζουν ποιος ολοκλήρωσε την διαδικασία ή ποιος δεν έλαβε καν μέρος σε αυτήν. Το πρόβλημα αυτό επιλύεται από το NRPS, επιτρέποντας στο εργαλείο να ζητήσει μια πλήρη λίστα μαθητών και καθηγητών χωρίς οι ίδιοι να έχουν εκκινήσει το εργαλείο. Έτσι οι εξεταστές μπορούν εύκολα να δουν ποιος έχει ολοκληρώσει την συγκεκριμένη δραστηριότητα.

3.4.2. Assignment and Grade Services

Πρόκειται για μια υπηρεσία του LTI Advantage που επιτρέπει στους καθηγητές να συγχρονίζουν τους βαθμούς των μαθητών μεταξύ εξωτερικών εργαλείων και συστημάτων εκμάθησης. Αυτή η υπηρεσία επιτρέπει, μαζί με τις βαθμολογίες των μαθητών να εμπεριέχονται και σχόλια από τον καθηγητή. Βασίζεται στην αντίστοιχη υπηρεσία της έκδοσης LTI 1.1, προσθέτοντας και την κατάσταση της εργασίας. Δηλαδή ο καθηγητής μπορεί να δει πότε ένας μαθητής έχει ξεκινήσει μια εργασία, και αν αυτή έχει ολοκληρωθεί.

3.4.3. Deep Linking

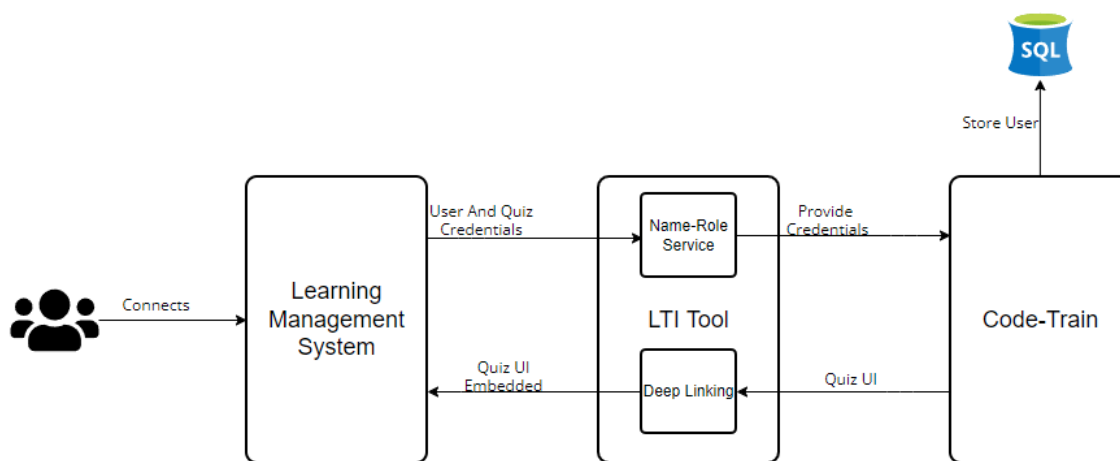
Η υπηρεσία Deep Linking επιτρέπει σε εξωτερικά εργαλεία να εμφανίζονται στην πλατφόρμα εκμάθησης σαν να είναι μέρος της. Στις παλαιότερες εκδόσεις το εργαλείο έπρεπε να ξεκινήσει ολόκληρο ακόμα και αν ο καθηγητής επιθυμούσε να εμφανίσει συγκεκριμένους πόρους από αυτό στους μαθητές. Το Deep Linking επιτρέπει στους καθηγητές να επιλέξουν τι περιεχόμενο θα εμφανίζεται στους μαθητές, αντί να τους αναγκάσει να εκκινήσουν το εργαλείο και στην συνέχεια να πλοηγηθούν στον επιθυμητό σύνδεσμο. (Global)



Εικόνα 5: Υπηρεσίες LTI Advantage

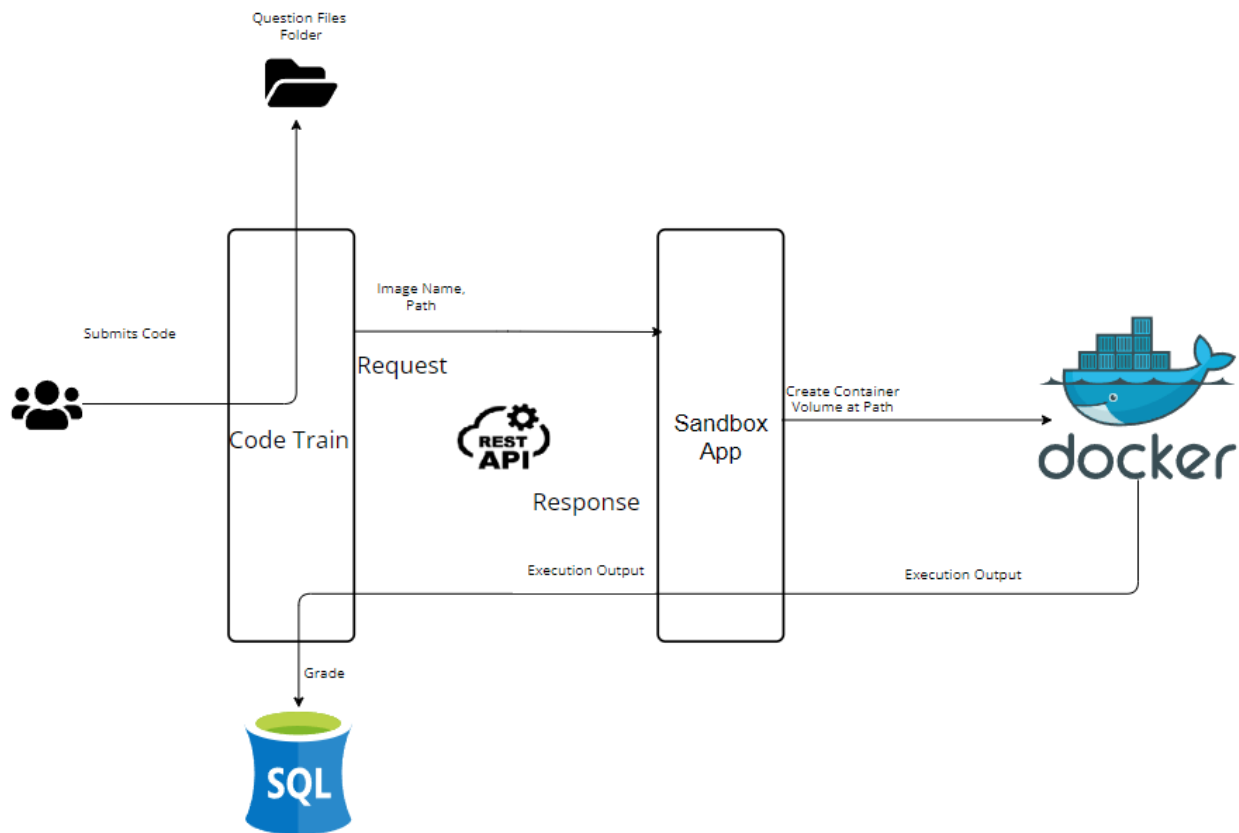
4. Περιγραφή Συστήματος

Το σύστημα συνολικά αποτελείται από τέσσερις εφαρμογές, οι οποίες αλληλοεπιδρούν μεταξύ τους κατά τη λειτουργία του. Οι εφαρμογές αυτές είναι το σύστημα διαχείρισης μάθησης(LMS) που χρησιμοποιείται στα πλαίσια του εκάστοτε μαθήματος, η εφαρμογή Code-Train, που χρησιμοποιείται για τη δημιουργία διαγωνισμάτων και την εξέταση των εκπαιδευόμενων, το LTI εργαλείο που καθιστά δυνατή την επικοινωνία του Code-Train με το εκάστοτε σύστημα διαχείρισης μάθησης, και τέλος την εφαρμογή sandbox που χρησιμοποιείται για την εκτέλεση του κώδικα που έχει υποβληθεί από τους εκπαιδευόμενους.



Εικόνα 6: Εγγραφή Χρήστη στο Code-Train μέσω Learning Management System

Αρχικά, ο εκπαιδευόμενος συνδέεται στο LMS και επιλέγει το διαγώνισμα για το οποίο θα αξιολογηθεί. Έπειτα, το LTI εργαλείο, μέσω της υπηρεσίας Name and Role service, μεταβιβάζει τα στοιχεία του εκπαιδευόμενου και τα στοιχεία του διαγωνίσματος στο Code-Train. Το Code-Train ελέγχει αν υπάρχει ο συγκεκριμένος εκπαιδευόμενος στη βάση, και αν δεν υπάρχει τον προσθέτει. Ακολούθως, ελέγχει στη βάση αν υπάρχει το διαγώνισμα με τα στοιχεία που έχουν μεταβιβαστεί. Αν το διαγώνισμα υπάρχει, τότε εμφανίζεται ενσωματωμένο στην σελίδα του LMS μέσω της υπηρεσίας deep-link του LTI εργαλείου. Διαφορετικά εμφανίζεται στον εκπαιδευόμενο μήνυμα λάθους.



Εικόνα 7: Διαδικασία υποβολής και βαθμολόγησης κώδικα

Αν ολοκληρωθεί επιτυχώς η διαδικασία επαλήθευσης του εκπαιδευόμενου και του διαγωνίσματος, τότε ξεκινάει η εξέταση. Μετά από κάθε υποβολή ερώτησης, ο κώδικας του εκπαιδευόμενου αποθηκεύεται σε συγκεκριμένο φάκελο μαζί με όλα τα αρχεία που χρειάζονται για την αξιολόγηση της κάθε ερώτησης. Το Code-Train μέσω restfull API, μεταβιβάζει το path αυτού του φακέλου στον δίσκο μαζί με το όνομα της εικόνας Docker που πρέπει να εκτελεστεί για τη βαθμολόγηση στην εφαρμογή Sandbox. Το sandbox δημιουργεί Container με volume στο path που έχει μεταβιβαστεί και εκτελεί την εικόνα μέσω Docker. Μόλις ολοκληρωθεί η εκτέλεση της εικόνας, το Sandbox επιστρέφει ως response την έξοδο της εκτέλεσης στο Code-Train, από όπου αποθηκεύεται στη βάση. Αυτή η διαδικασία ακολουθείται για κάθε ερώτηση του διαγωνίσματος.

4.1. Code-Train

Πρόκειται για την εφαρμογή με χαρακτηριστικά συστήματος διαχείρισης μάθησης, που αφορά εκπαιδευτές της επιστήμης των υπολογιστών. Οι κύριες λειτουργίες της εφαρμογής αποτελούν η δημιουργία Διαγωνίσματος, Προφίλ Ερώτησης, Ερώτησης καθώς και η βαθμολόγηση των διαγωνισμάτων.

4.1.1. Λειτουργίες Code-Train

4.1.1.1. Δημιουργία Προφίλ Ερώτησης

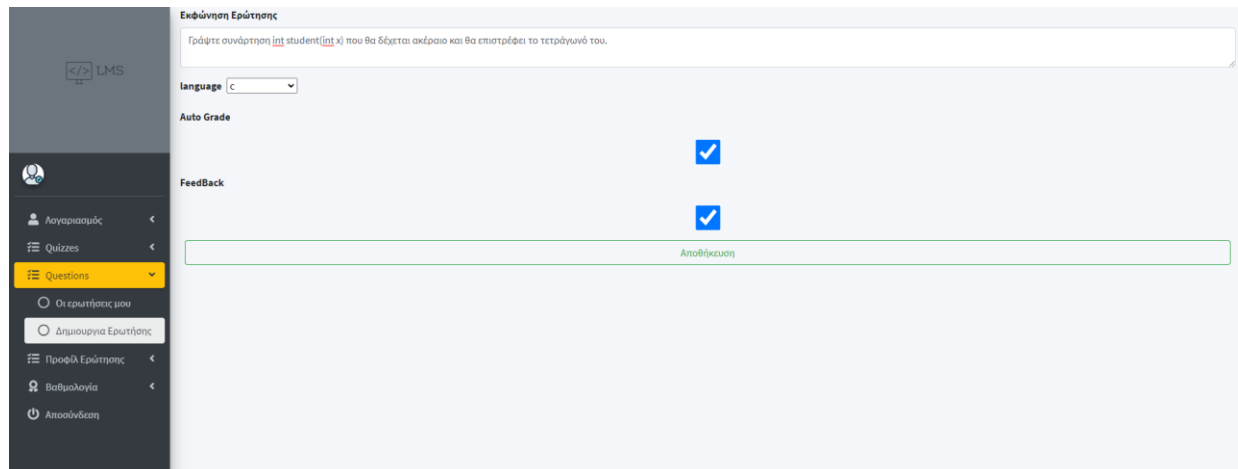
Το Προφίλ Ερώτησης είναι ένα πρότυπο πάνω στο οποίο βασίζονται οι ερωτήσεις της εφαρμογής και καθορίζει τον τρόπο βαθμολόγησης της ερώτησης που το χρησιμοποιεί. Δημιουργούνται από χρήστες της εφαρμογής, αποθηκεύονται στη βιβλιοθήκη του συστήματος και μπορούν να χρησιμοποιηθούν από όλους τους χρήστες, ανεξαρτήτως του δημιουργού τους. Κατά τη δημιουργία τους ζητώνται μια αναλυτική περιγραφή του προφίλ και των απαιτήσεών του, μια εικόνα docker ή ένα dockerfile, καθώς υπάρχει και επιλογή για την επισύναψη ενδεχόμενων αρχείων που χρειάζονται για την εκτέλεση της εικόνας docker. Τέλος υπάρχουν πεδία για την επιλογή της γλώσσας προγραμματισμού, τον τρόπο βαθμολόγησης, δηλαδή αν πρόκειται για αυτόματη ή χειροκίνητη βαθμολόγηση καθώς και για το αν ο εξεταζόμενος λάβει ανατροφοδότηση σχετικά με τα αποτελέσματά του μετά το τέλος της εξέτασης. Στην περίπτωση που ο δημιουργός επισυνάψει dockerfile, τότε το δημιουργείται εικόνα με βάση αυτό το αρχείο, διαφορετικά, χρησιμοποιείται η εικόνα docker που έχει επισυναπτεί.

The screenshot shows the 'Δημιουργία Προφίλ Ερώτησης' (Create Question Profile) form. On the left is a dark sidebar with the LMS logo and a menu containing: Λογαριασμός, Quizzes, Questions, Προφίλ Ερώτησης (highlighted), Συλλογή, Δημιουργία Προφίλ, Βαθμολογία, and Αποσύνδεση. The main form area has a title 'Δημιουργία Προφίλ Ερώτησης' and a subtitle 'Περιγραφή'. It contains a large text area for a description with a placeholder text in JSON format. Below this are sections for 'Docker Image' (with a 'Choose File' button), 'Multiple Files' (with a 'Choose Files' button), 'Docker File' (with a 'Choose File' button), 'Language' (a dropdown menu set to 'c'), 'Auto Grade' (a checked checkbox), and 'Feedback' (a checked checkbox). A 'Submit' button is at the bottom right.

Εικόνα 8: Δημιουργία Προφίλ Ερώτησης

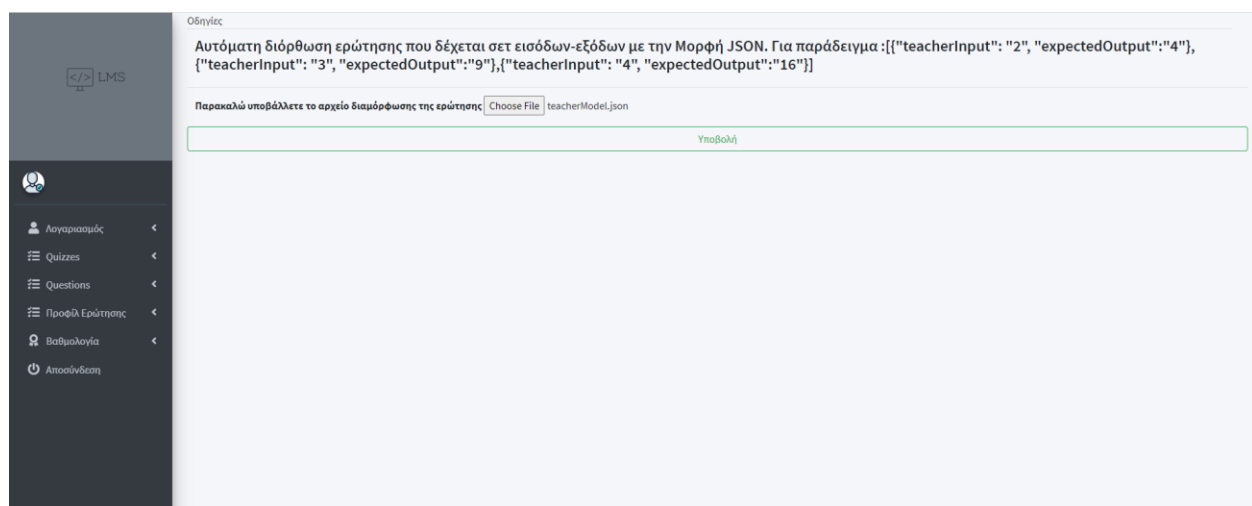
4.1.1.1. Δημιουργία Ερώτησης

Ο καθηγητής δημιουργεί τις ερωτήσεις και αποθηκεύονται στη βιβλιοθήκη του συστήματος ώστε να υπάρχει δυνατότητα χρήσης τους σε περισσότερα από ένα διαγωνίσματα. Κατά τη δημιουργία ερωτήσεων ζητώνται η εκφώνησή της, η γλώσσα προγραμματισμού που υποστηρίζει, ο τρόπος βαθμολόγησής της και η επιλογή παροχής ανατροφοδότησης στους εξεταζόμενους μετά τη λήξη της εξέτασης.



Εικόνα 9: Δημιουργία Ερώτησης

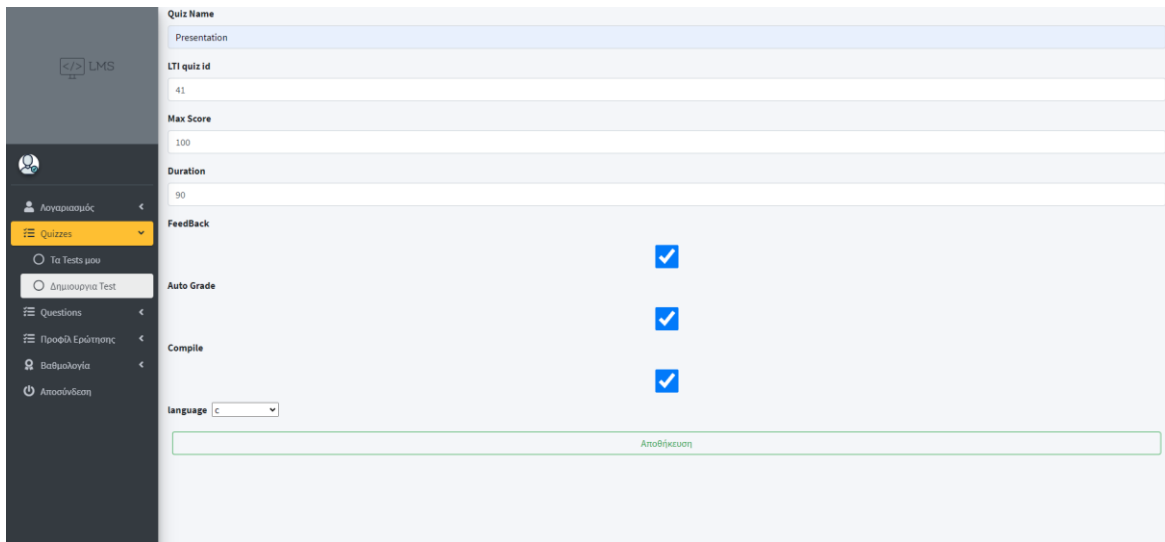
Αφού διαμορφωθεί η ερώτηση, και ανάλογα με τις επιλογές διαμόρφωσής της, εμφανίζεται στον δημιουργό μια λίστα από τα προφίλ που συνάδουν στις επιλογές αυτές, δηλαδή υποστηρίζουν την ίδια γλώσσα προγραμματισμού, τον ίδιο τρόπο βαθμολόγησης και την ίδια επιλογή ανατροφοδότησης. Έπειτα, σύμφωνα με τις οδηγίες του προφίλ ερώτησης που θα επιλεγεί, ο δημιουργός καλείται να επισυνάψει ένα αρχείο JSON που να καλύπτει της απαιτήσεις του προφίλ.



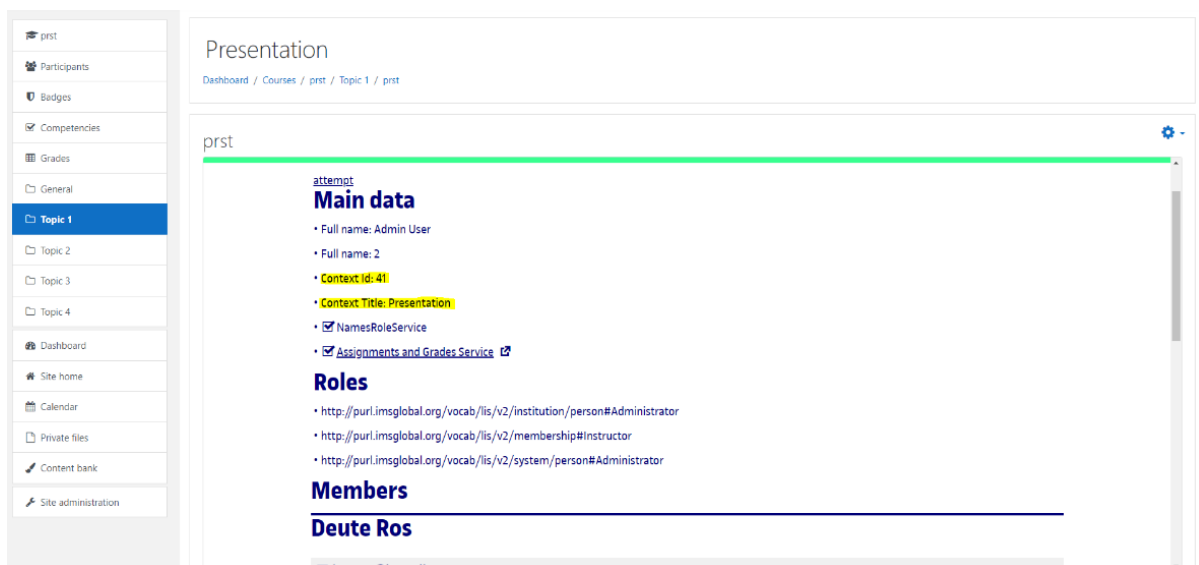
Εικόνα 10: Επισύναξη JSON αρχείου Διαμόρφωσης

4.1.1.1. Δημιουργία Διαγωνίσματος

Για τη δημιουργία ερώτησης, ο καθηγητής συμπληρώνει το όνομα του διαγωνίσματος, τη μέγιστη βαθμολογία που μπορούν να λάβουν οι εξεταζόμενοι, τη διάρκειά του, καθώς και τις επιλογές διαμόρφωσης για τη γλώσσα προγραμματισμού, τον τρόπο βαθμολόγησης και την επιλογή ανατροφοδότησης στους εξεταζόμενους. Επίσης στη διαμόρφωση του διαγωνίσματος προστίθεται η επιλογή για το εάν επιτρέπεται το compile του κώδικα από τους εξεταζόμενους. Τέλος υπάρχει το πεδίο LTI quiz id το οποίο χρησιμοποιείται για την αντιστοίχιση του διαγωνίσματος με κάποιο σε Σύστημα Διαχείρισης Μάθησης. Το LTI id είναι ένα από τα πεδία του διαγωνίσματος που εξάγονται από το LTI εργαλείο και μεταφέρονται στο Code Train ώστε να επιτευχθεί η αντιστοίχιση.



Εικόνα 11: Δημιουργία Διαγωνίσματος



Εικόνα 12: Εξαγωγή Στοιχείων Διαγωνίσματος Από LMS μέσω LTI εργαλείου

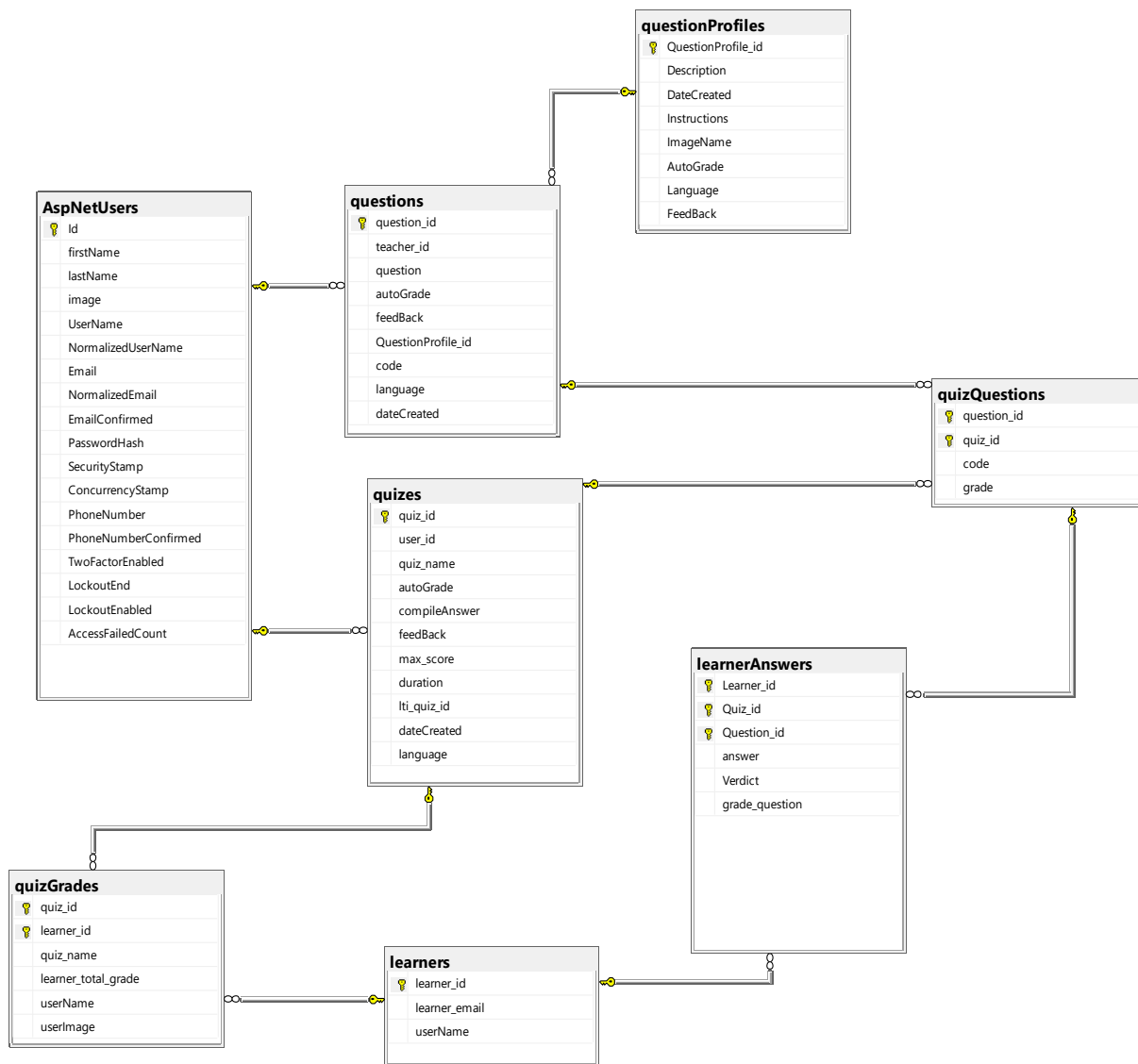
Μετά την ολοκλήρωση της διαμόρφωσης του διαγωνίσματος, ο δημιουργός επιλέγει τις ερωτήσεις που επιθυμεί να συμπεριλάβει, και που συνάδουν με τις επιλογές διαμόρφωσης του, και τέλος επισυνάπτει την βαθμολογία της κάθε μίας.

Εκφώνηση	Βαθμός
Συνάρτηση $\text{student}(\text{int } x)$ που θα δέχεται ακέραιο και θα επιστρέφει το τετράγωνό του.	30
Συνάρτηση $\text{student}(\text{int } x)$ που θα δέχεται ακέραιο και θα επιστρέφει τον κύβο του.	40
Συνάρτηση $\text{student}(\text{int } x)$ που θα δέχεται την μία πλευρά ενός τετραγώνου και θα επιστρέφει την περίμετρό του.	30

Αποθήκευση

Εικόνα 13: Εισαγωγή Βαθμολογίας Ερωτήσεων

4.1.1. Βάση Δεδομένων Code-Train



Εικόνα 14: Βάση Δεδομένων Code-Train

Ο πίνακας AspNetUsers αναφέρεται στους χρήστες της εφαρμογής που δημιουργούν λογαριασμό και συνδέονται σε αυτή. Κάθε χρήστης δημιουργεί διαγωνίσματα quizzes και ερωτήσεις questions. Οι δύο αυτοί πίνακες έχουν ως ξένο κλειδί το user_id και teacher_id αντίστοιχα που είναι το Id του χρήστη που τα δημιούργησε. Έχουν μόνο έναν χρήστη ως δημιουργό, ενώ ο χρήστης διατηρεί πολλές ερωτήσεις και διαγωνίσματα. Επίσης ο πίνακας questions έχει ξένο κλειδί το QuestionProfile_id που δείχνει στον πίνακα questionProfiles, δηλαδή στο προφίλ ερώτησης βάσει του οποίου έχει δημιουργηθεί. Κάθε ερώτηση έχει ένα μόνο προφίλ ερώτησης, ενώ κάθε προφίλ ερώτησης μπορεί να χρησιμοποιηθεί σε παραπάνω από μία ερωτήσεις. Οι ερωτήσεις και τα διαγωνίσματα συνδέονται μέσω του πίνακα quizQuestions, ο οποίος υποδηλώνει τις ερωτήσεις που βρίσκονται σε ένα διαγώνισμα, και έχει ως πρωτεύον κλειδί, το συνδυασμό των quiz_id και question_id από τους πίνακες quizzes και questions αντίστοιχα. Κάθε φορά που κάποιος χρήστης πλατφόρμας εκμάθησης συνδέεται στο Code-Train, δημιουργείται μια εγγραφή στον πίνακα learners. Τα στοιχεία τους εισάγονται αυτόματα στη βάση από την πλατφόρμα και δημιουργείται ένας μοναδικός κωδικός για κάθε χρήστη που είναι το πρωτεύον κλειδί του πίνακα. Ο πίνακας learnerAnswers είναι η απάντηση που δίνει ο αξιολογούμενος για κάθε ερώτηση στο εκάστοτε διαγώνισμα. Το πρωτεύον κλειδί αυτού του πίνακα είναι ο συνδυασμός των κλειδιών Learner_id, Quiz_id και Question_id από τους πίνακες learners, quizzes και questions αντίστοιχα. Τα υπόλοιπα πεδία του πίνακα είναι η απάντηση που έδωσε ο αξιολογούμενος, ο βαθμός που έλαβε και η έξοδος του compiler για την κάθε ερώτηση του διαγωνίσματος. Τέλος, ο αξιολογούμενος συνδέεται με το Διαγώνισμα στο οποίο λαμβάνει μέρος μέσω του πίνακα quizGrades. Ο πίνακας αυτός έχει ως κλειδί τον συνδυασμό των κλειδιών των πινάκων learners και quizzes και περιέχει τον συνολικό βαθμό που έλαβε ο αξιολογούμενος στο διαγώνισμα.

4.2. Εφαρμογή Sandbox

Η εφαρμογή Sandbox υλοποιήθηκε χρησιμοποιώντας Spring Boot και το πακέτο spring boot starter web. Η κύρια λειτουργία που υποστηρίζει η εφαρμογή Sandbox είναι η εκτέλεση εικόνων docker. Δέχεται post request με παράμετρο όνομα της εικόνας που θα εκτελεστεί και ένα string που αντιπροσωπεύει ένα path στο σύστημα. Το Sandbox δημιουργεί ένα Container χρησιμοποιώντας volume που δείχνει σε αυτό το path και εκτελεί την εικόνα με την εντολή docker run. Μόλις διευθετηθεί η εκτέλεση της εικόνας, το Container τερματίζεται, και επιστρέφεται η έξοδος της εκτέλεσης. Η εφαρμογή sandbox επίσης υποστηρίζει τη λειτουργία δημιουργίας εικόνας docker. Δέχεται ένα post request με ένα αρχείο dockerfile και δημιουργεί την εικόνα χρησιμοποιώντας την εντολή docker image build. Αν η εικόνα δημιουργηθεί επιτυχώς, επιστρέφεται το όνομά της, ενώ σε αντίθετη περίπτωση επιστρέφεται μήνυμα λάθους.

4.3 Επικοινωνία Sandbox - Code-Train

Το Sandbox επικοινωνεί με την .Net εφαρμογή και αποτελεί το ασφαλές απομονωμένο περιβάλλον εκτέλεσης κώδικα των εξεταζόμενων. Ο εξεταζόμενος υποβάλει τον κώδικά του στην .Net εφαρμογή από την πλατφόρμα Moodle με τη χρήση του Lti tool. Έπειτα, δημιουργείται ένας μοναδικός φάκελος για κάθε εξεταζόμενο που περιέχει τον υποβαλλόμενο κώδικα σε αρχείο, το json αρχείο που έχει υποβάλει ο καθηγητής για την συγκεκριμένη ερώτηση και όλα τα υπόλοιπα αρχεία που έχουν υποβληθεί και είναι απαραίτητα για το συγκεκριμένο προφίλ ερώτησης. Η .Net εφαρμογή υποβάλει ένα post request στο Sandbox στέλνοντας το όνομα της docker εικόνας του συγκεκριμένου προφίλ ερώτησης καθώς και το path που έχει ο φάκελος των αρχείων στο σύστημα. Το sandbox εκτελεί τον κώδικα που βρίσκεται στον φάκελο με το συγκεκριμένο Path και επιστρέφει την έξοδο στην .Net εφαρμογή.

Επίσης, οι δύο εφαρμογές αλληλοεπιδρούν κατά τη δημιουργία προφίλ ερώτησης στην .Net εφαρμογή, όταν υποβάλλεται αρχείο dockerfile αντί για Docker Image. Η .Net εφαρμογή υποβάλει post request στέλνοντας το dockerfile και λαμβάνει το όνομα της εικόνας που δημιουργείται, αποθηκεύοντάς το στη βάση ώστε να χρησιμοποιηθεί στην εκτέλεση κώδικα για κάθε ερώτηση που χρησιμοποιεί το συγκεκριμένο προφίλ.

5. Υλοποίηση Συστήματος

Η εφαρμογές που υλοποιήθηκαν στα πλαίσια της διπλωματικής εργασίας είναι το Code-Train, με χρήση του framework ASP.NET και της γλώσσας προγραμματισμού C#, καθώς και η εφαρμογή Sandbox, με χρήση του framework Spring Boot σε Java. Για το LTI εργαλείο χρησιμοποιήθηκε το εργαλείο java-lti-1.3 provider example (Xavi Aracil, 2018) που υλοποιήθηκε στο ίδρυμα University of Catalonia. Για σύστημα διαχείρισης μάθησης χρησιμοποιήθηκε η πλατφόρμα Moodle, ωστόσο μπορεί να χρησιμοποιηθεί οποιοδήποτε σύστημα υποστηρίζει την τεχνολογία LTI Advantage.

5.1. Code-Train

Η εφαρμογή Code-Train υλοποιήθηκε με το αρχιτεκτονικό μοντέλο asp.net core mvc. Για τη δημιουργία χρηστών χρησιμοποιήθηκε το identity framework, ενώ για την σχεδίαση της βάσης δεδομένων το entity framework. Υλοποιήθηκαν τέσσερις controllers, οι Profile, Home, Quiz και Question.

5.1.1. Δημιουργία Χρηστών και αλληλεπίδραση μέσω LTI

Οι χρήστες δημιουργούνται στον Profile Controller, όπου επίσης επεξεργάζονται τα στοιχεία τους. Ο Home controller είναι υπεύθυνος για την επικοινωνία του Code-Train και του συστήματος διαχείρισης μάθησης μέσω του εργαλείου LTI. Αυτό επιτυγχάνεται με αίτημα του εργαλείου στο Code-Train που περιέχει το username, το email, και τον ρόλο του συνδεδεμένου χρήστη, καθώς και ενός ακεραίου id που επιτρέπει την ταυτοποίηση του εκάστοτε διαγωνίσματος. Έπειτα ελέγχεται ο ρόλος του χρήστη. Αν είναι εκπαιδευτής, τότε επιστρέφεται μια σελίδα που περιέχει μια λίστα με όλους τους μαθητές που έλαβαν μέρος στο διαγώνισμα με την βαθμολογία τους. Αν είναι εκπαιδευόμενος, τότε ελέγχεται αν ο συγκεκριμένος χρήστης είναι εγγεγραμμένος στη βάση δεδομένων του συστήματος, και αν δεν είναι αποθηκεύεται. Μόλις ολοκληρωθεί η διαδικασία ταυτοποίησης, τότε επιστρέφεται η σελίδα του διαγωνίσματος.

5.1.2. Ερωτήσεις και Προφίλ Ερωτήσεων

Ο Question Controller είναι υπεύθυνος για την δημιουργία των ερωτήσεων από τους εκπαιδευτές, καθώς και για τα Προφίλ πάνω στα οποία θα είναι βασισμένες. Η διαδικασία δημιουργίας ερώτησης και επιλογής Προφίλ Ερώτησης είναι τετριμμένη, ωστόσο αξίζει να αναφερθεί η διαδικασία δημιουργίας Προφίλ Ερώτησης. Κατά τη δημιουργία Προφίλ Ερώτησης, μεταξύ άλλων ζητείται η υποβολή αρχείου εικόνας Docker ή Dockerfile. Έπειτα ελέγχεται ποια από τις δύο επιλογές αρχείων έχει υποβληθεί. Στην περίπτωση που έχει υποβληθεί εικόνα Docker, τότε εξάγεται το όνομά της και αποθηκεύεται στο κατάλληλο πεδίο στη βάση. Διαφορετικά, γίνεται ένα post request στη Sandbox εφαρμογή που περιέχει το αρχείο Dockerfile ως παράμετρο. Το Sandbox δημιουργεί εικόνα Docker από το Dockerfile, την αποθηκεύει και επιστρέφει ως response το όνομά της, και αυτό αποθηκεύεται στη βάση όπως και στην περίπτωση υποβολής εικόνας Docker. Τέλος, όλα τα αρχεία που έχουν υποβληθεί αποθηκεύονται στο

σύστημα, ώστε να χρησιμοποιηθούν στην υποβολή ερωτήσεων από τους εκπαιδευόμενους κατά την εξέτασή τους.

5.1.3. Δημιουργία και Εξέταση Διαγωνίσματος

Οι ενέργειες που αφορούν την δημιουργία και την διεξαγωγή των διαγωνισμάτων εκτυλίσσονται στον Quiz Controller. Αφότου ο εκπαιδευτής συμπληρώσει τις μεταβλητές στο γραφικό περιβάλλον δημιουργίας διαγωνίσματος, εμφανίζονται σε λίστα όλες από τις ερωτήσεις που έχει δημιουργήσει ο ίδιος, έχουν την ίδια γλώσσα προγραμματισμού. Μόλις επιλέξει όποιες από αυτές επιθυμεί να εισάγει στο διαγώνισμα, δημιουργείται στη βάση δεδομένων ένα ζεύγος ερώτησης-διαγωνίσματος που αντιπροσωπεύει τις ερωτήσεις που περιέχονται σε ένα διαγώνισμα. Αφού ολοκληρωθεί η παραπάνω διαδικασία, ο εκπαιδευτής θέτει το ποσοστό της βαθμολογίας που θα έχει η κάθε ερώτηση από τη συνολική βαθμολογία του διαγωνίσματος.

Κατά την υποβολή απάντησης σε μια ερώτηση διαγωνίσματος, δημιουργείται μοναδικός φάκελος για κάθε εκπαιδευόμενο, και η απάντησή του αποθηκεύεται στον συγκεκριμένο φάκελο ως αρχείο. Έπειτα συλλέγονται όλα τα αρχεία που έχουν υποβληθεί κατά τη δημιουργία του προφίλ πάνω στο οποίο βασίζεται η ερώτηση και αποθηκεύονται επίσης στον συγκεκριμένο φάκελο. Πραγματοποιείται έπειτα ένα post request στη sandbox εφαρμογή όπου εκτελείται ο κώδικας, με παραμέτρους το όνομα της εικόνας Docker του προφίλ ερώτησης, και το Path του φακέλου στον δίσκο. Ως απάντηση λαμβάνεται η έξοδος της εκτέλεσης, και εξάγονται τα αποτελέσματα της αξιολόγησης, δηλαδή ο βαθμός, αν η ερώτηση βαθμολογείται αυτόματα, η έξοδος της εκτέλεσης του κώδικα του εκπαιδευόμενου καθώς και ενδεχόμενα σχόλια για την εκτέλεση που έχει θέσει ο εκπαιδευτής, τα οποία εμφανίζονται στον εκπαιδευόμενο όταν ολοκληρωθεί το διαγώνισμα.

5.2. Εφαρμογή Sandbox

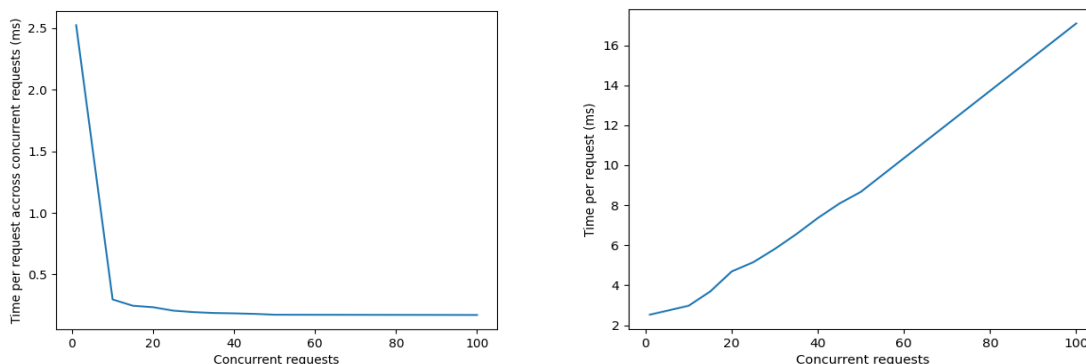
Η εφαρμογή Sandbox αναπτύχθηκε χρησιμοποιώντας Spring Boot σε γλώσσα Java, χρησιμοποιώντας το πακέτο Spring Boot Web για τη διαχείριση Rest requests. Οι λειτουργίες της εφαρμογής είναι η δημιουργία εικόνων Docker και η εκτέλεσή τους.

Η δημιουργία εικόνας Docker γίνεται στον Docker controller. Ο controller αυτός δέχεται αίτημα που περιέχει ένα αρχείο dockerfile. Το αρχείο αυτό αποθηκεύεται προσωρινά σε φάκελο στο σύστημα. Η εικόνα δημιουργείται με την εντολή docker image build με ορίσματα το Path του φακέλου όπου έχει αποθηκευτεί το dockerfile και του ονόματος της εικόνας που βασίζεται στην ημερομηνία και την ώρα του αιτήματος, ώστε να διασφαλιστεί η μοναδικότητα του ονόματός της. Αν ολοκληρωθεί επιτυχώς η διαδικασία, διαγράφεται ο φάκελος που περιέχει το dockerfile και επιστρέφεται το όνομα της εικόνας που έχει δημιουργηθεί. Σε διαφορετική περίπτωση επιστρέφεται μήνυμα σφάλματος.

Ο Compiler Controller είναι υπεύθυνος για την εκτέλεση εικόνων Docker. Δέχεται αίτημα το οποίο περιέχει το όνομα της εικόνας που θα εκτελεστεί, μαζί με ένα path στον δίσκο. Δημιουργείται Container με volume σε αυτό το path και αποθηκεύεται το αποτέλεσμα της εκτέλεσης της εικόνας. Μόλις ολοκληρωθεί η εκτέλεση της εικόνας, επιστρέφεται το αποτέλεσμα της.

6. Αξιολόγηση του Code-Train

Σκοπός αυτής της ενότητας είναι η αξιολόγηση της απόδοσης της εφαρμογής Code-Train κατά τη λήψη και την διεκπεραίωση αιτημάτων με τη χρήση του Apache Bench. Το Apache Bench είναι ένα εργαλείο αξιολόγησης διακομιστών, κατακλύζοντάς τους με HTTP requests και καταγράφοντας μετρήσεις για τους χρόνους απόκρισης και την επιτυχία αυτών των αιτημάτων. Η αξιολόγηση διεξάγεται μέσα από γραμμή εντολών και οι παράμετροι που εισάγονται είναι ο συνολικός αριθμός των αιτημάτων, ο σύνδεσμος στον οποίο θα διεξαχθούν αυτά τα αιτήματα καθώς και ο αριθμός των πολλαπλών αιτημάτων που θα διεξαχθούν παράλληλα στο διακομιστή. Διεξήχθησαν 10000 αιτήματα αυξάνοντας κάθε φορά τον αριθμό των παράλληλων αιτημάτων. Στον σύνδεσμο στον οποίο έγιναν τα αιτήματα υπάρχει αλληλεπίδραση με τη βάση δεδομένων, ώστε να διασφαλιστεί η αξιοπιστία και η εγκυρότητα της αξιολόγησης. Οι μετρήσεις που πάρθηκαν αφορούν τον μέσο χρόνο που απαιτείται για την επεξεργασία τόσο ενός αιτήματος (time per request), όσο και για ένα παράλληλο σύνολο αιτημάτων (time across concurrent requests). Τα αποτελέσματα των μετρήσεων φαίνονται στις ακόλουθες γραφικές παραστάσεις.



Διαγράμματα 1&2: Αξιολόγηση απόδοσης Code-Train

Παρατηρούμε πως στην περίπτωση της επεξεργασίας συνόλου παράλληλων αιτημάτων, όσο αυξάνονται τα αιτήματα, ο χρόνος επεξεργασίας μειώνεται. Αυτό συμβαίνει διότι με την αύξηση των παράλληλων αιτημάτων, τα συνολικά αιτήματα στον server μειώνονται, με αποτέλεσμα να απαιτείται όλο και λιγότερος χρόνος για την επεξεργασία τους. Ωστόσο, ο χρόνος αυτός σταθεροποιείται στα 0.17 ms στέλνοντας 50 ή περισσότερα παράλληλα αιτήματα στο server. Αντιθέτως, όσο αυξάνεται ο αριθμός των αιτημάτων που στέλνονται παράλληλα, είναι λογικό να αυξάνεται και ο χρόνος που απαιτείται για την επεξεργασία του καθενός από αυτά. Στη γραφική παράσταση φαίνεται πως ο χρόνος επεξεργασίας αυξάνεται αναλογικά με την αύξηση του αριθμού των αιτημάτων που στέλνονται παράλληλα στο server.

7. Συμπεράσματα

Στα πλαίσια αυτής της διπλωματικής εργασίας αναπτύχθηκε ένα σύστημα που αφορά την αυτόματη αξιολόγηση ερωτήσεων προγραμματισμού, παρέχοντας παράλληλα στους εκπαιδευτές την δυνατότητα να οργανώνουν τα διαγωνίσματά τους.

7.1. Περίληψη

Η εξ αποστάσεως εκπαίδευση και η online μάθηση έχει αναπτυχθεί τα τελευταία χρόνια. Οι εκπαιδευτές έχουν στη διάθεσή τους πληθώρα εργαλείων που βοηθούν στην οργάνωση και την διεξαγωγή των μαθημάτων και των εξετάσεων τους. Το σύστημα της προκειμένης διπλωματικής επικεντρώνεται στα μαθήματα προγραμματισμού. Παρέχει στους εκπαιδευτές τη δυνατότητα να διατηρούν αρχείο με τα διαγωνίσματα και τις ερωτήσεις που έχουν δημιουργήσει καθώς και με τις βαθμολογίες των φοιτητών τους.

Η κύρια λειτουργία του συστήματος είναι η επικοινωνία με πλατφόρμες εκμάθησης, όπως το Moodle, και η αυτόματη αξιολόγηση ερωτήσεων κώδικα σε ασφαλές και απομονωμένο περιβάλλον. Η εκτέλεση και η αξιολόγηση του κώδικα που έχει υποβληθεί από τους εκπαιδευόμενους επιτυγχάνεται με την εκτέλεση εικόνων Docker. Το σημαντικό πλεονέκτημα που παρέχεται στους εκπαιδευτές, είναι η πλήρης ευελιξία στον τρόπο με τον οποίο γίνεται η αξιολόγηση, καθώς οι εικόνες Docker που εκτελούνται υποβάλλονται κατά τη δημιουργία των ερωτήσεων.

7.2. Μελλοντική Επέκταση

Όσον αφορά τη μελλοντική επέκταση, το σύστημα μπορεί να επεκταθεί σε μεγάλο βαθμό. Αρχικά, το σύστημα θα λειτουργεί ως αυτόνομη πλατφόρμα εκμάθησης, και η σύνδεσή του με άλλες πλατφόρμες μέσω LTI θα είναι απλώς μια λειτουργία. Θα υποστηρίζεται ο ρόλος των εκπαιδευόμενων, οι οποίοι θα εγγράφονται στην πλατφόρμα, όπως και οι εκπαιδευτές. Η διεξαγωγή των διαγωνισμάτων θα γίνεται μέσα απευθείας στην εφαρμογή, και οι εκπαιδευόμενοι θα κρατούν αρχείο με τις βαθμολογίες τους. Τέλος προβλέπεται να υποστηρίζονται επιπλέον τύποι ερωτήσεων για τα διαγωνίσματα, όπως πολλαπλής επιλογής, ή ανάπτυξης κειμένου.

Βιβλιογραφία

- Ellis, R. K. (2009). A Field Guide to Learning Management Systems. *Learning Circuits*. Ανάκτηση από [1]https://web.archive.org/web/20140824102458/http://www.astd.org/~media/Files/Publications/LMS_fieldguide_20091
- Wasik, S. (2017). A Survey on Online Judge Systems and Their Applications. Ανάκτηση από <https://arxiv.org/abs/1710.05913>
- HN, N. P. (2021). Building a secure/sandboxed environment for executing untrusted code. Ανάκτηση από <https://dev.to/narasimha1997/building-a-secure-sandboxed-environment-for-executing-untrusted-code-7e8>
- Richard Lobb, U. o. (2022). The Architecture of CodeRunner. Ανάκτηση από https://github.com/trampgeek/moodle-qtype_coderunner#the-architecture-of-coderunner
- Baeldung. (χ.χ.). Introduction to Servlets and Servlet Containers. Ανάκτηση από <https://www.baeldung.com/java-servlets-containers-intro>
- Baeldung. (χ.χ.). What is a Spring Bean? Ανάκτηση από <https://www.baeldung.com/spring-bean>
- Docs, D. (χ.χ.). Docker overview. Ανάκτηση από <https://docs.docker.com/get-started/overview/>
- ElearningLessonsGreece. (-). Εκπαιδευτική πλατφόρμα του Moodle. Ανάκτηση από <https://elearninglessonsgreece.wordpress.com/%CE%B5%CE%BA%CF%80%CE%B1%CE%B9%CE%B4%CE%B5%CF%85%CF%84%CE%B9%CE%BA%CE%AE-%CF%80%CE%BB%CE%B1%CF%84%CF%86%CF%8C%CF%81%CE%BC%CE%B1-%CF%84%CE%BF%CF%85-moodle/>
- Global, I. (χ.χ.). IMS LTI® 1.3 and LTI Advantage. Ανάκτηση από <https://www.imsglobal.org/activity/learning-tools-interoperability>
- Jaap Eldering, N. G. (2004). DOMjudge. Ανάκτηση από <https://www.domjudge.org/about>
- Lobb, R. (2015). Jobe: REST API. Ανάκτηση από <https://github.com/trampgeek/jobeblob/master/restapi.pdf>
- Lobb, R. (2022). Ανάκτηση από <https://github.com/trampgeek/jobeblob/master/restapi.pdf>
- Razi, M. (χ.χ.). Interaction-within-MVC-pattern. Ανάκτηση από <https://www.researchgate.net/profile/Mjm-Razi-2/publication/328716094/figure/fig2/AS:688713864597504@1541213589083/Interaction-within-MVC-pattern-The-Model-component-correlates-with-all-the-data-related.ppm>
- rj45. (2008). Simple Example of MVC (Model View Controller) Design Pattern for Abstraction. Ανάκτηση από <https://www.codeproject.com/Articles/25057/Simple-Example-of-MVC-Model-View-Controller-Design>
- Sletton, E. (2020). What is REST API? An Overview. Ανάκτηση από <https://www.liquidweb.com/kb/what-is-rest-api-an-overview/>

Smith, S. (2022). Overview of ASP.NET Core MVC. Ανάκτηση από <https://docs.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-6.0>

Tutorial, E. F. (χ.χ.). What is Entity Framework? Ανάκτηση από <https://www.entityframeworktutorial.net/what-is-entityframework.aspx>

TutorialsPoint. (χ.χ.). Spring Boot - Introduction. Ανάκτηση από https://www.tutorialspoint.com/spring_boot/spring_boot_introduction.htm

Xavi Aracil, U. O. (2018). java-lti-1.3-provider-example. Ανάκτηση από <https://github.com/UOC/java-lti-1.3-provider-example>