



ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

**Σχεδιασμός και ανάπτυξη διαδικτυακής εφαρμογής για τη
διάχυση πληροφορίας σχετικά με εγκαταστάσεις
πολιτισμικής κληρονομιάς/σημασίας**

Παππάς Χριστόδουλος
Επιβλέπων: Ιωάννης Καρύδης

Νοέμβριος 2020

Επιβλέπων

Ιωάννης Καρύδης, Αναπληρωτής Καθηγητής,
Ιόνιο Πανεπιστήμιο, Τμήμα Πληροφορικής

Τριμελής Επιτροπή

Ιωάννης Καρύδης, Αναπληρωτής Καθηγητής
Ιόνιο Πανεπιστήμιο, Τμήμα Πληροφορικής

Θεόδωρος Ανδρόνικος, Επίκουρος Καθηγητής,
Ιόνιο Πανεπιστήμιο, Τμήμα Πληροφορικής

Μιχαήλ Στεφανιδάκης, Επίκουρος Καθηγητής,
Ιόνιο Πανεπιστήμιο, Τμήμα Πληροφορικής

“If I have seen further it is by standing on the shoulders of Giants.”

-Isaac Newton

Περίληψη

Η παρούσα πτυχιακή εργασία αναλύει την εφαρμογή διαδικασιών για τη δημιουργία μιας ολοκληρωμένης διαδικτυακής εφαρμογής που έχει ως σκοπό τη διάχυση πληροφορίας για εγκαταστάσεις πολιτισμικής κληρονομιάς.

Γίνεται ανάλυση του πώς από την αποθήκευση και διάχυση πληροφορίας μέσω του πάπυρου και των βιβλίων φτάσαμε στις μέρες μας να έχουμε πρόσβαση σε απεριόριστες πηγές πληροφόρησης χάρη στη χρήση του διαδικτύου. Και πώς η μετάδοση μηνυμάτων πλέον μπορεί να γίνει με την χρήση των Qr Codes και τι άλλες τεχνολογίες χρησιμοποιούνται σήμερα.

Έπειτα, αναλύεται ο τρόπος με τον οποίο δημιουργήθηκε αυτή η εφαρμογή. Ποιες γλώσσες προγραμματισμού χρησιμοποιήθηκαν, ποια αρχιτεκτονικά σχέδια λογισμικού ακολουθήθηκαν και γιατί είναι σημαντικό να ακολουθούνται κατά την υλοποίηση μιας εφαρμογής. Επιπρόσθετα, γίνεται αναφορά στο πώς πρέπει να γίνει μια σωστή εγκατάσταση διακομιστή και πώς μπορεί να επωφεληθεί από αυτό μια εφαρμογή. Παράλληλα, γίνεται ανάλυση του Public API που δημιουργήθηκε στα πλαίσια της πτυχιακής εργασίας και περιγράφεται πώς μπορούν οι χρήστες να δημιουργήσουν τη δική τους εφαρμογή χρησιμοποιώντας τα δεδομένα του QBI με ασφαλή τρόπο.

Τέλος, παρουσιάζονται τα αποτελέσματα ενός ερωτηματολογίου, που απαντήθηκε από μια μερίδα ατόμων. Το ερωτηματολόγιο περιείχε ερωτήσεις σχετικά με τις λειτουργίες και την ευκολία χρήσης της εφαρμογής. Σκοπός του ερωτηματολογίου ήταν η έμπρακτη χρήση της εφαρμογής από πραγματικούς χρήστες, η αξιολόγηση του περιεχομένου και η παρουσίαση της μετάβασης ενός χρήστη από αναγνώστη σε δημιουργό και ποιοι κίνδυνοι υπάρχουν σε αυτό.

Πρόλογος

Το QBI (Qr Building Information) πρόκειται για ένα λογισμικό το οποίο προσφέρεται στο χρήστη δωρεάν και αποσκοπεί στη διάχυση πληροφορίας σχετικά με κτίρια και περιοχές πολιτισμικής και κληρονομικής σημασίας. Το εαρινό εξάμηνο του 2015, στα πλαίσια εργασίας εξαμήνου για το μάθημα «Πολιτισμική Πληροφορική» ο Ανδρέας Αποστόλου και εγώ αποφασίσαμε να αναλάβουμε από κοινού την υλοποίηση ενός website όπου θα περιέχει πληροφορίες για τα κτίρια που απαρτίζουν το Ιόνιο Πανεπιστήμιο.

Για κάθε σελίδα δημιουργήθηκε ένα μοναδικό Qr Code το οποίο γινόταν να εκτυπωθεί και να πλαστικοποιηθεί ώστε να πλαισιωθεί έξω από κάθε κτίριο.

Παρέχοντας έτσι εύκολη και γρήγορη πρόσβαση στον οποιοδήποτε ενδιαφέρεται να μάθει για την πλούσια ιστορία αυτών των κτιρίων και τη χρήση τους μέχρι σήμερα.

Με την καθοδήγηση και επίβλεψη του καθηγητή, Ιωάννη Καρύδη, καταφέραμε να φέρουμε εις πέρας αυτή την εργασία, ενώ ταυτόχρονα για πρώτη φορά, τρία τμήματα του Ιονίου Πανεπιστημίου συνεργάστηκαν ώστε να υπάρξει ένα καλό αποτέλεσμα.

Πιο συγκεκριμένα, από το τμήμα Μετάφρασης και Διερμηνείας του Ιονίου Πανεπιστημίου και τις φοιτήτριες Φούκα Σταυρούλα και Λυγούρα Ευαγγελία με επιβλέποντα καθηγητή τον κύριο Κεραμιδά Σωτήριο, έγινε η μετάφραση των κειμένων της ιστοσελίδας. Τελευταία συνεργασία είναι από το τμήμα Τεχνών Ήχου και Εικόνας και τον φοιτητή Παναγιώτη Καλογύρου ο οποίος επιμελήθηκε την φωτογράφιση των κτιρίων.

Η συγκεκριμένη εργασία είχε βαθμολογηθεί με άριστα.

Έπειτα από την επιτυχία αυτής της εργασίας αποφασίσαμε πως θα ήταν εφικτή η επέκταση αυτής της υπηρεσίας σε ένα Software as a Service. Όπου ο οποιοσδήποτε θα μπορεί να συνδεθεί με έναν λογαριασμό και να δημιουργήσει το δικό του κτίριο το οποίο θα είναι θεατό από όλους και κάνοντας χρήση της λειτουργίας δημιουργίας μοναδικού Qr Code μέσα από την υπηρεσία θα μπορεί να το εκτυπώσει και να το πλαισιώσει έξω από το κτίριό του. Η δεύτερη έκδοση του εν λόγω έργου είναι αυτή η οποία περιγράφεται παρακάτω.

Ιωάννινα, Νοέμβριος 2020,
Χριστόδουλος Παππάς

Περιεχόμενα

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ.....	1
Επιβλέπων	2
Τριμελής Επιτροπή	2
Περίληψη	4
Πρόλογος	6
Περιεχόμενα.....	7
Εισαγωγή	9
Βασικές ιδέες	10
Διάχυση πληροφορίας	10
Από τον πάπυρο στα Βιβλία	10
Η γέννηση του διαδικτύου	10
Μετάδοση πληροφορίας μέσω σχημάτων (Qr Code)	12
Βιβλιογραφική έρευνα	13
Εισαγωγή	13
Διάχυση Πληροφορίας Μέσω Ιστοσελίδας	13
Πολιτισμική Κληρονομιά.....	14
Qr Code (Ετικέτα γρήγορης ανταπόκρισης)	15
Συμπεράσματα	15
Υλοποίηση έργου.....	17
Εισαγωγή	17
Επιλογή εργαλείων.....	17
Βάση δεδομένων.....	17
Ένωση βάσης δεδομένων με backend	20
Ανάλυση λειτουργιών Public Api	23
Model.....	27
View	28
View-Model.....	28
Προγραμματισμός προχωρημένων Λειτουργιών QBI.....	28
Δημιουργία εγκατάστασης πολιτισμικής κληρονομιάς/σημασίας από το διαχειριστικό.....	31
Εγκατάσταση διακομιστή και δημιουργία ονόματος	34

Συμπεράσματα.....	38
Εισαγωγή	38
Αναλυτικά τα αποτελέσματα	38
Ιδέες επέκτασης.....	45
CI/CD (Συνεχής Ενσωμάτωση/Συνεχής ανάπτυξη)	45
IoT (Beacon, Proximity).....	45
Εμπειρία 3D	46
Βιβλιογραφία.....	47
Γλωσσάρι Ξενικών Όρων.....	48

Εισαγωγή

Συχνά, ταξιδεύοντας σε άλλες χώρες ή άλλες πόλεις της Ελλάδας μπορεί να δω ένα κτίριο ή ένα μνημείο το οποίο θα μου κεντρίσει το ενδιαφέρον χωρίς όμως να έχω εύκολη πρόσβαση σε ιστορικά γεγονότα ή άλλες ενδιαφέρουσες πληροφορίες. Όπως εγώ, έτσι και άλλοι άνθρωποι θεωρώ πως έχουν την ίδια ανάγκη και δίψα για γνώση. Οι παραπάνω λόγοι είναι και αυτοί που με ώθησαν στη δημιουργία αυτής της υπηρεσίας.

Η συγκεκριμένη υπηρεσία η οποία διατίθεται δωρεάν (<https://theqbi.com>) προσφέρει στο χρήστη ή οργανισμό την εύκολη δημιουργία ιστοσελίδας, παρέχοντας τη δυνατότητα να προσθέσει τις πληροφορίες που μπορεί κάποιος άλλος χρήστης να χρειάζεται.

Οι χρήστες πλέον μπορούν να δημιουργήσουν τη δική τους ιστοσελίδα σχετικά εύκολα. Όμως ταυτόχρονα θα πρέπει να δαπανήσουν αρκετό χρόνο στην εύρεση και πληρωμή ονόματος, την ενοικίαση διακομιστή, ακόμη και την αγορά κάποιου template ώστε η ιστοσελίδα του να είναι ευανάγνωστη από κινητή πλατφόρμα. Επίσης μεγάλο κόστος είναι και η μίσθωση συνεργάτη για την υλοποίηση και τη συντήρηση της ιστοσελίδας.

Με τη χρήση της υπηρεσίας QBI όλα τα παραπάνω εμπόδια εξαλείφονται μιας και το μόνο που χρειάζεται να κάνουν οι χρήστες είναι να συμπληρώσουν τις πληροφορίες που απαιτούνται.

Επιπλέον έχοντας δημιουργήσει χρήστη στην υπηρεσία παρέχεται πρόσβαση στο δωρεάν API (Application Programming Interface) όπου οι χρήστες μπορούν να χρησιμοποιήσουν τα δεδομένα που ήδη υπάρχουν στην πλατφόρμα για τη δημιουργία ιστοσελίδας ή ακόμη και εφαρμογής για κινητά.

English Version

Often, while traveling to other countries or Greek cities I come across interesting sites or monuments without having any way to get more information about what I see. I believe that other people also seek knowledge like I do. The aforementioned reasons pushed me into creating this service.

QBI is a web application available for free (<https://theqbi.com>) that offers a user or an organization an easy way of creating a website. Giving the opportunity to add information that another user on the internet might need.

Everyone can create a website relatively easily. But they will have to overcome a series of impediments, like finding and paying for an available domain name, paying for hosting, buying a fancy html template in order for his website to be legible from a mobile platform. It might also be required to hire a software developer for the implementation and maintenance of the website. But, by using the QBI service all that has to be done is just fill in the information needed. In addition, after creating a user at the QBI, they will be granted access to a free public API (Application Programming Interface) where they can use the data that are already available on the platform to create a web page or even a mobile application.

Βασικές ιδέες

Διάχυση πληροφορίας

Είναι η μετάδοση πληροφορίας ενός μηνύματος χωρίς την άμεση ανταπόκριση από το κοινό. Βασίζεται στην παραδοσιακή θεωρία της επικοινωνίας, που απαρτίζεται από έναν αποστολέα και έναν παραλήπτη. Ο ένας στέλνει την πληροφορία και ο δεύτερος την λαμβάνει, την επεξεργάζεται και στέλνει πληροφορίες πίσω, όπως γίνεται ακόμη και σήμερα με τις τηλεφωνικές γραμμές. Σε αυτή την περίπτωση ο αποστολέας στέλνει τις πληροφορίες κάνοντας τις διαθέσιμες όχι σε ένα άτομο αλλά σε πολλά.

Από τον πάπυρο στα Βιβλία

Από τα αρχαία χρόνια ο άνθρωπος είχε καταλάβει την σημαντικότητα της αποθήκευσης και μετάδοσης πληροφορίας, για την διατήρηση της ιστορικής κληρονομιάς και του πολιτισμού. Ένας από τους αρχαιότερους τρόπους μετάδοσης και αποθήκευσης πληροφορίας είναι ο πάπυρος. Είναι ένα φυτό των τελμάτων πολύ συχνό στην αρχαιότητα αλλά δυστυχώς υπό εξαφάνιση σήμερα. Μέσω μιας πολύπλοκης διαδικασίας μετατρέπονταν στο γνωστό μέχρι σήμερα πάπυρο. Ο πάπυρος ήταν ο πιο κοινός τρόπος γραφής στην αρχαιότητα μέχρι την επικράτηση της περγαμηνής. Χρησιμοποιήθηκε περίπου από το 3000 π.χ. μέχρι τον 10^ο αιώνα. Είναι αρκετά εύθραυστος σαν υλικό και έπρεπε να διατηρείται μόνο σε ξηρά μέρη που δεν έρχονται σε επαφή με τον αέρα και το ισχυρό φως. Μετά τον πάπυρο και την περγαμηνή, σειρά στον έντυπο λόγο έχουν τα βιβλία. Στο μεσαίωνα τα βιβλία ήταν κυρίως χειρόγραφα. Περιείχαν θρησκευτικό περιεχόμενο αλλά και μερικά αρχαία έργα και είναι και ο τρόπος με τον οποίο αυτά διασώθηκαν μέχρι τις μέρες μας. Τα πρώτα τυπωμένα βιβλία εμφανίστηκαν τον 6^ο αιώνα μ.Χ.. Η μεγαλύτερη άνθηση στην παραγωγή βιβλίου έρχεται στη βιομηχανική επανάσταση (19^{ος} – 20^{ος} αιώνας). Με την μηχανοποίηση και τον αυτοματισμό στα τυπογραφεία, το να παραχθούν εκατοντάδες βιβλία έπαψε να είναι δύσκολο και οικονομικά ασύμφορο. Δημιουργήθηκαν αμέτρητες βιβλιοθήκες από κράτη, ιδρύματα, ιδιώτες και πολιτισμικούς οργανισμούς απαθανατίζοντας με αυτό τον τρόπο την πολιτισμική κληρονομιά, τα ήθη και έθιμα του κάθε τόπου και την ιστορία (19-20).

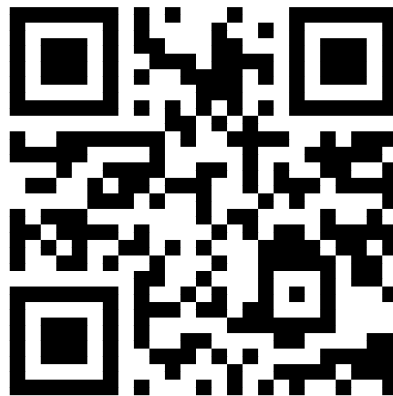
Η γέννηση του διαδικτύου

Ένα από τα πιο σημαντικά επιτεύγματα του ανθρώπου για τη διάχυση πληροφορίας μετά το βιβλίο, είναι το web 1.0 (ιστός) ή World Wide Web γνωστό και ως www (Παγκόσμιος ιστός). Πρόκειται για ένα σύστημα πληροφορίας όπου αρχεία γίνονται προσιτά από όλο τον κόσμο με τη χρήση ενός λογισμικού που ονομάζεται web browser (φυλλομετρητής). Όλα ξεκίνησαν το 1989 από τον Άγγλο επιστήμονα Tim Berners-Lee, ο οποίος όντας εργαζόμενος στο CERN του ήταν πολύ δύσκολο να βρει πληροφορίες οι οποίες ήταν αποθηκευμένες σε διαφορετικούς υπολογιστές. Το 1980 υπέβαλε ένα υπόμνημα για ένα λογισμικό που είχε κατασκευάσει σχετικά με ένα σύστημα διαχείρισης πληροφορίας μεταξύ υπολογιστών χρησιμοποιώντας κόμβους και υπερσυνδέσμους. Η πρώτη έκδοση του web περιείχε πληροφορία σε μορφή hypertext (υπερκειμένου) που μπορούσαν να παρέχουν μόνο όσοι γνώριζαν HTML (Hypertext Markup Language). Άρα η χρήση του διαδικτύου ήταν κυρίως η παθητική λήψη πληροφοριών. Με την έλευση του web 2.0 αυτό άλλαξε, μιας και προσέφερε έναν πιο δραστικό τρόπο διάχυσης πληροφορίας. Βασίζεται στις αρχές της διάδρασης ανθρώπου και υπολογιστή. Ο χρήστης μπορεί όχι μόνο να αλλάξει το περιβάλλον μιας

ιστοσελίδας αλλά να παρέμβει και στο περιεχόμενό της χωρίς να χρειάζεται να γνωρίζει web development. Οι πιο γνωστές εφαρμογές που ήρθαν με τη δημιουργία του web 2.0 είναι τα κοινωνικά δίκτυα, τα blog και οι εφαρμογές πολυμεσικού περιεχομένου (όπως φωτογραφία, βίντεο και ήχος) (21).

Μετάδοση πληροφορίας μέσω σχημάτων (Qr Code)

Στις ημέρες μας η πληροφορία είναι διαθέσιμη στο κοινό από οπουδήποτε και σε οποιαδήποτε μορφή. Είτε αυτή είναι απλό κείμενο είτε κωδικοποιημένη. Μια μορφή κωδικοποίησης πληροφορίας είναι το γνωστό σε όλους μας barcode. Μια φωτογραφία με μαύρες και άσπρες γραμμές οι οποίες απεικονίζουν μια σειρά αριθμών (11). Μια μεταγενέστερη μορφή κωδικοποίησης πληροφορίας είναι αυτή του Qr Code (Quick Response code). Πρόκειται για έναν διδιάστατο γραμμικό κώδικα αναγνώσιμο από την κάμερα του κινητού μας τηλεφώνου με την χρήση λογισμικού και περιέχει πληροφορίες σχετικά με το αντικείμενο στο οποίο είναι προσαρτημένο. Τα πλεονεκτήματά του από το barcode είναι ότι μπορεί να περιέχει περισσότερη πληροφορία αλλά και άλλους χαρακτήρες πέρα από αριθμούς. Μια από τις χρήσεις τους είναι και η εύκολη πρόσβαση σε ιστοσελίδες και είναι η βασική ιδέα αυτής της εργασίας. (13)



Εικόνα 1: Qr Code για την εύκολη μετάβαση κάποιου στο QBI και στη σελίδα που παρέχει πληροφορίες για την Ιόνιο Ακαδημία.

Βιβλιογραφική έρευνα

Εισαγωγή

Σε αυτήν την ενότητα θα περιγράψω πώς ενώθηκαν διάφορες τεχνολογίες και έρευνες προκειμένου να έχω ως αποτέλεσμα την εφαρμογή QBI. Από το πόσο σημαντική είναι στις μέρες μας η διάχυση πληροφορίας και σε ποιους κλάδους εφαρμόζεται, μέχρι το πώς η πολιτισμική κληρονομιά αποθηκεύεται και μεταδίδεται σε άλλους ανθρώπους και πολιτισμούς. Τέλος, θα αναλύσω την τεχνολογία των Qr Code και πώς αυτά βοηθάνε τέτοιου είδους εφαρμογές.

Διάχυση Πληροφορίας Μέσω Ιστοσελίδας

Το διαδίκτυο αποτελεί πηγή απεριόριστης πληροφορίας για πληθυσμούς σε όλο τον κόσμο, για αυτόν τον λόγο η σωστή σχεδίαση μιας ιστοσελίδας έχει σημαντικές θεωρητικές και πρακτικές επιπτώσεις. Είναι απαραίτητο να ακολουθείται ένας μηχανισμός για οργάνωση, αποθήκευση και πρόσβαση σε πληροφορίες από τους χρήστες. Για να επιτευχθεί αυτό υπάρχουν τα Web portals τα οποία είναι σχεδιασμένα ως SOA (Service Oriented Architecture) (9). Η υλοποίησή τους είναι συνήθως σε κάποια γλώσσα προγραμματισμού Web όπως HTML, CSS, JavaScript και με την βοήθειά τους δημιουργούνται ιστοσελίδες οι οποίες προσφέρουν στο χρήστη πληροφορίες έχοντας μια μορφοποίηση. Έχουν άμεση αλληλεπίδραση με τον χρήστη και διαρκώς παρέχουν νέα πληροφορία. Τέτοιες ιστοσελίδες μπορεί να περιέχουν άρθρα, επιστημονικές εργασίες, ανακοινώσεις, εργαστήρια αλλά και πληροφορίες σχετικά με εγκαταστάσεις πολιτισμικής κληρονομιάς.

Με τα κινητά τηλέφωνα, η διάχυση πληροφορίας μέσω ιστοσελίδων γίνεται ολοένα και πιο δημοφιλής. Πλέον οι χρήστες δεν χρειάζεται να βρίσκονται μπροστά από κάποιον υπολογιστή προκειμένου να βρουν κάτι που ψάχνουν. Μπορούν απλά να αναζητήσουν οτιδήποτε μέσω της κινητής τους συσκευής (9). Για αυτό, και η σωστή σχεδίαση μιας εφαρμογής είναι εξαιρετικά σημαντική στις μέρες μας, ειδικά όταν πρόκειται για κινητές συσκευές. Ένας χρήστης μπορεί να φύγει από μια ιστοσελίδα αν δεν βρει αυτό που ψάχνει αρκετά γρήγορα. Πρέπει τα περιεχόμενα να είναι ευανάγνωστα και σωστά δομημένα, προκειμένου κάποιος χρήστης να βρει αμέσως αυτό που ψάχνει. Ακόμη και ο χρόνος φόρτωσης του περιεχομένου παίζει πολύ σημαντικό λόγο στο αν κάποιος χρήστης που επισκέπτεται την ιστοσελίδα μας θα μείνει σε αυτή. Η διάρκεια προσοχής που έχει ο μέσος άνθρωπος είναι μόλις 8 δευτερόλεπτα και ολοένα και μειώνεται. Για αυτό πρέπει να υλοποιηθούν τεχνικές οι οποίες που πρέπει να ακολουθούνται κατά την υλοποίηση μιας ιστοσελίδας.

Πολιτισμική Κληρονομιά

Το παρελθόν ενός πολιτισμού ή τόπου αποτελεί μια άγνωστη γη. (12) Ωστόσο κάθε λαός πρέπει να μαθαίνει για αυτή και το ιστορικό της υπόβαθρο. Οι γρήγοροι ρυθμοί εξέλιξης που βιώνουμε σήμερα οφείλονται σε ότι έχουμε διδαχθεί από τους προγόνους μας και ότι έχει καταγραφεί. Η πολιτισμική κληρονομιά, μπορεί να είναι υλική είτε άυλη. Από ένα έθιμο το οποίο αναβιώνετε από ομάδες ατόμων μέχρι ένα αρχαίο αντικείμενο το οποίο κεντρίζει το ενδιαφέρον αυτών που το αντικρίζουν. Η προστασία της πολιτιστικής κληρονομιάς ενός πολιτισμού συνίσταται στην ανεύρεση, την συνεχή καταγραφή, την μελέτη και την συντήρηση και αποκατάσταση των υλικών και άυλων αγαθών ενός τόπου. Η κύρια επιδίωξη είναι όχι μόνο η διάσωση και η συντήρηση της πολιτισμικής κληρονομιάς, αλλά και η διάχυση αυτής ώστε να γίνει προσιτή και να εκτιμηθεί από τον λαό κάθε πολιτισμού. Με αυτόν τον τρόπο, μπορεί κάποιος να συνειδητοποιήσει και να εκτιμήσει τα ιδιαίτερα χαρακτηριστικά της πολιτιστικής ταυτότητας ενός πολιτισμού.

Επί σειρά ετών, ο άνθρωπος αποθήκευε την πολιτιστική κληρονομιά με την χρήση των βιβλίων. Ήταν ένας τρόπος διατήρησης και διάχυσης των χαρακτηριστικών και ιδιαιτεροτήτων ενός λαού. Στις μέρες μας με την χρήση του διαδικτύου, η αποθήκευση και διάχυση πληροφορίας έχει γίνει πολύ εύκολη. Και έτσι πολλοί οργανισμοί σπεύδουν να δημιουργήσουν ιστοσελίδες στο διαδίκτυο που περιέχουν πληροφορίες για διάφορα πολιτισμικά αγαθά.

Τα δεδομένα πολιτιστικής κληρονομιάς είναι συντακτικά και σημασιολογικά ετερογενή, πολύγλωσσα και πλούσια σημασιολογικά. Στις μέρες μας η πληροφορία αυτή παράγεται από μουσεία, βιβλιοθήκες, αρχεία και οργανισμούς πολυμέσων, αλλά και από μεμονωμένα άτομα κάνοντας χρήση διαδικτυακών υπηρεσιών όπως αυτή που περιγράφεται στην παρούσα εργασία (12). Η εφαρμογή των αρχών και των τεχνολογιών του Σημασιολογικού Ιστού είναι μια νέα, πολλά υποσχόμενη προσέγγιση για την αποθήκευση της πολιτιστικής κληρονομιάς στο διαδίκτυο.

Qr Code (Ετικέτα γρήγορης ανταπόκρισης)

Πρόκειται για έναν γραμμωτό κώδικα δύο διαστάσεων, του οποίου η σάρωση γίνεται με την χρήση ενός κινητού τηλεφώνου. Την περασμένη δεκαετία σημειώθηκε σταθερή αύξηση της χρήσης αυτών των κωδικών Qr με την έλευση των έξυπνων κινητών τηλεφώνων. Ωστόσο η χρήση των Qr Code στο πεδίο της πολιτισμικής κληρονομιάς βρίσκεται ακόμη σε αρχικό στάδιο (11).

Σε έναν κώδικα Qr μπορούν να αποθηκευτούν πολλές πληροφορίες, όπως η διεύθυνση μιας ιστοσελίδας ή απλά κάποιο αναγνωριστικό (Για παράδειγμα το αναγνωριστικό ενός μνημείου). Η χρήση του Qr Code γίνεται όλο και πιο διαδεδομένη σε μουσεία, ιστορικές πόλεις και αρχαιολογικούς χώρους. Επιτρέπει στους χρήστες την εύκολη πρόσβαση σε πληροφορία μέσω συγκεκριμένων ιστοσελίδων. Η τοποθέτηση του Qr Code δεν προκαλεί πρόβλημα σε κάποιο μνημείο ή κάποιο έκθεμα, και μπορεί πολύ εύκολα να αντικατασταθεί αν διαβρωθεί. Επιπρόσθετα οι χρήστες πλέον έχουν εξοικειωθεί αρκετά με την χρήση του Qr Code κάνοντας την πλοήγηση σε διάφορες ιστοσελίδες αρκετά εύκολη (11-12).

Συμπεράσματα

Η παρούσα πτυχιακή εργασία έχει ως σκοπό την ένωση των τριών παραπάνω θεμάτων σε ένα. Με την χρήση μιας διαδικτυακής πλατφόρμας γίνεται εύκολο για κάποιον χρήστη ή οργανισμό να αποθηκεύσει και να διανέμει πληροφορίες σχετικά με την κουλτούρα και τον πολιτισμό με τρόπο εύκολο και γρήγορο για έναν χρήστη. Με την χρήση των κωδικών Qr Code ένας χρήστης ο οποίος απλά περνάει έξω από μια εγκατάσταση πολιτισμικής αξίας μπορεί να σαρώνει με το έξυπνο τηλέφωνό του και να διαβάσει περισσότερες πληροφορίες.

Υλοποίηση έργου

Εισαγωγή

Ένα web application (διαδικτυακή εφαρμογή) πρόκειται για λογισμικό το οποίο «τρέχει» σε έναν διακομιστή και είναι προσπελάσιμο από οποιονδήποτε έχει πρόσβαση στο διαδίκτυο. Σε αντίθεση με το λογισμικό υπολογιστών, ένα web application δεν χρειάζεται να εγκατασταθεί σε κάποιο λειτουργικό σύστημα αλλά αρκεί η χρήση ενός web browser (φυλομετρητής). Είναι σχεδιασμένο με βάση το μοντέλο client-server (χρήστης-διακομιστής) και παρέχει συγκεκριμένες υπηρεσίες. Μια τέτοια υπηρεσία μπορεί να είναι το email, social media (κοινωνικά δίκτυα) κτλ.

Επιλογή εργαλείων

Για τη δημιουργία ενός web application πριν την υλοποίηση είναι πολύ σημαντικό να γίνουν μερικά απλά βήματα.

1. Ορισμός του προβλήματος προς επίλυση.
2. Σχεδιασμός ροής της εφαρμογής.
3. Δημιουργία πρωτότυπου της εμφάνισης της εφαρμογής.
4. Επικύρωση του πρωτοτύπου από πιθανούς χρήστες.
5. Επιλογή των σωστών εργαλείων για την υλοποίηση.

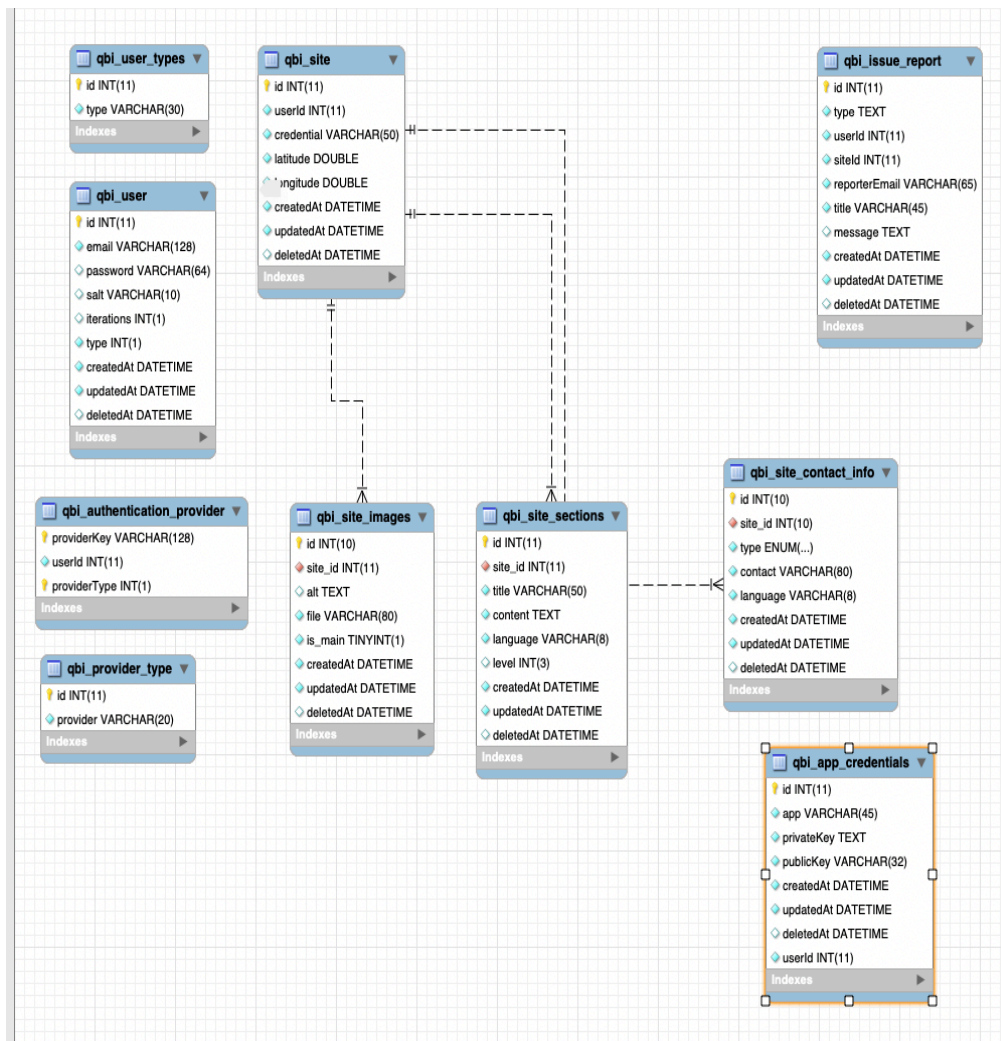
Η επιλογή σωστών εργαλείων είναι από τα πιο σημαντικά βήματα στη δημιουργία μιας εφαρμογής. Πρέπει να είναι κάτι το οποίο ταιριάζει σε αυτό που θέλουμε να δημιουργήσουμε και να μην επιλέξουμε κάτι απλά επειδή είναι δημοφιλές. Για παράδειγμα, για τη δημιουργία μιας απλής εφαρμογής διάχυσης πληροφορίας που περιέχει μια και μόνο σελίδα θα ήταν υπερβολικό να χρησιμοποιήσουμε Django σε συνδυασμό με Angular.

Για τη δημιουργία του QBI χρησιμοποίησα mysql για τη βάση δεδομένων, Angular για τη δημιουργία του frontend και nodejs με TypeScript για τη δημιουργία του backend.

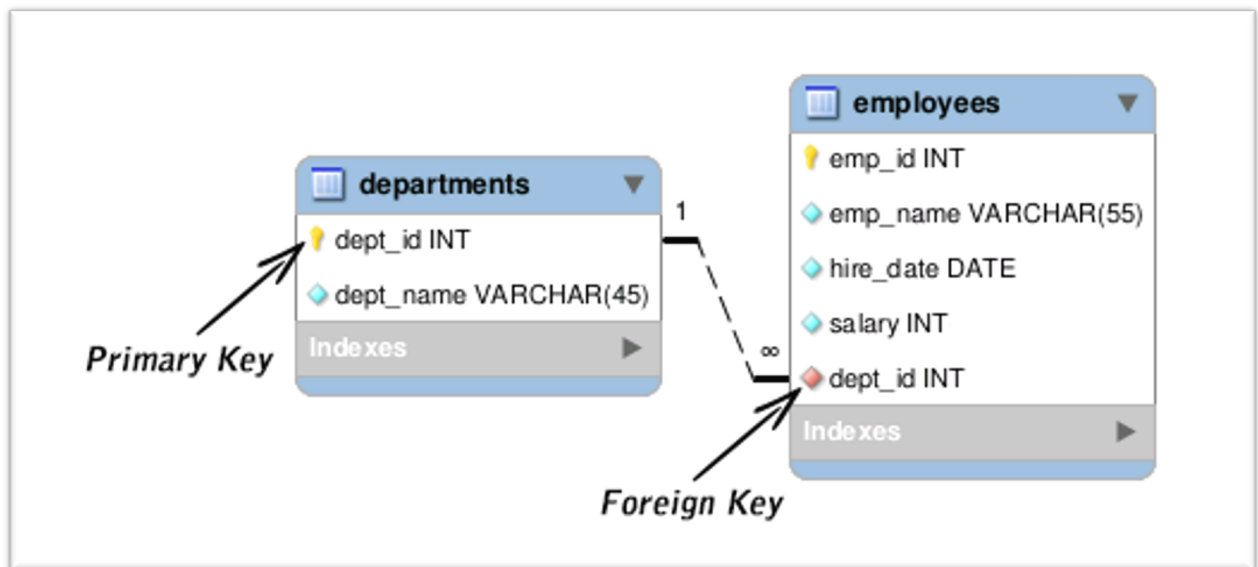
Προκειμένου να μπορώ να υλοποιώ την εφαρμογή από οποιοδήποτε υπολογιστή θέλω και είναι πιο εύκολη τελικά η εγκατάσταση όλων των κομματιών στο διακομιστή επέλεξα να χρησιμοποιήσω GIT. Το GIT είναι ένα ανοιχτού κώδικα και δωρεάν σύστημα το οποίο βοηθά στην κατανομή του κώδικα κρατώντας ιστορικό και έχοντας αμέτρητες λειτουργίες βοηθά στην διευκόλυνση κάποιου προγραμματιστή ή ακόμη και ομάδων.

Βάση δεδομένων

Η βάση δεδομένων, από τα πιο σημαντικά κομμάτια κατά τη διάρκεια της υλοποίησης της μιας εφαρμογής, είναι μια συλλογή από μορφοποιημένα σχετιζόμενα δεδομένα στα οποία είναι δυνατή η αποθήκευση, ανανέωση και ανάκτηση δεδομένων μέσω αναζήτησης κατ' απαίτηση. Υπάρχουν πολλοί τύποι βάσεων δεδομένων, όπως σχεσιακές βάσεις δεδομένων και no-sql βάσεις δεδομένων. Για την συγκεκριμένη εργασία επέλεξα να χρησιμοποιήσω τη σχεσιακή βάση δεδομένων mysql.



Αρχικά, σε ένα απλό χαρτί έγραψα τους πίνακες και πρόσθεσα τα δεδομένα που θα ήθελα αυτοί οι πίνακες να κρατάνε. Έπειτα σημείωσα ποιοι από αυτούς τους πίνακες έπρεπε να έχουν συσχετίσεις μεταξύ τους μέσω foreign key (ξένο κλειδί). Για παράδειγμα στη συγκεκριμένη εφαρμογή ήθελα ο πίνακας με τους χρήστες και ο πίνακας που περιέχει τα κτίρια να έχουν συσχέτιση μεταξύ τους μέσω του αναγνωριστικού του χρήστη.



Για να είναι λειτουργική η βάση δεδομένων και να σιγουρευτώ πως η υλοποίηση θα είναι σταθερή πριν μεταβεί σε κάποιον διακομιστή έπρεπε να εγκαταστήσω τα απαραίτητα προγράμματα τοπικά και να ξεκινήσω την υλοποίηση. Αφού εγκατέστησα τα απαραίτητα εργαλεία ([mysql server](#), mysql workbench) πέρασα το σχήμα της βάσης από το χαρτί στον τοπικό διακομιστή μέσω εντολών mysql αλλά και του προγράμματος διεπαφής χρήστη mysql workbench.

Στο τέλος, και αφού όλη η υλοποίηση της εφαρμογής είχε τελειώσει με όλες τις λειτουργίες της έπρεπε μεταφέρω τη βάση δεδομένων σε άλλο διακομιστή ώστε να μπορέσει να ενωθεί με τη διεπαφή αποθήκευσης, ανάκτησης και επεξεργασίας δεδομένων την οποία εξηγώ παρακάτω.

Ο διακομιστής που χρησιμοποίησα για την εφαρμογή παρέχεται από την [digitalocean](#) όπου παρέχει κοινόχρηστη φιλοξενία μέσω μηχανημάτων με διακομιστή Linux. Ο κύριος λόγος που επέλεξα αυτή την υπηρεσία είναι διότι προσφέρει απόλυτο έλεγχο στον προγραμματιστή και έτσι μπορούσα πιο εύκολα να πειραματιστώ.

Ο συγκεκριμένος διακομιστής δεν παρέχει κάποια διεπαφή χρήστη για να έχει καλύτερη απόδοση χωρίς να χρειάζεται το μηχάνημα να καταναλώνει επεξεργαστική ισχύ για τη δημιουργία γραφικών. Έτσι έπρεπε να κάνω όλη την εγκατάσταση μέσω εντολών. Για να συνδεθώ στον εξωτερικό διακομιστή χρησιμοποίησα το τερματικό του υπολογιστή μου και εκτελώντας την εντολή **ssh root@ip διακομιστή** αποκτούσα πλήρη έλεγχο στο διακομιστή. Έπειτα εγκατέστησα την mysql καλώντας τις παρακάτω εντολές. (16)

```
sudo apt update
sudo apt install mysql-server
sudo mysql_secure_installation
```

Η πρώτη εντολή ανανεώνει τον δείκτη πακέτων στο διακομιστή ώστε να λάβουμε την νεότερη έκδοση από το πρόγραμμα που θέλουμε να εγκαταστήσουμε. Η επόμενη εντολή εγκαθιστά τον διακομιστή mysql.

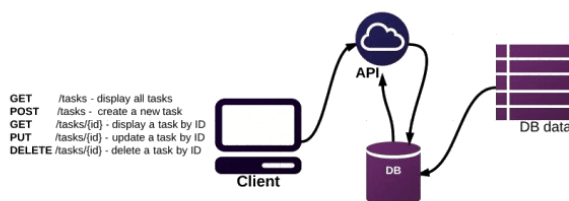
Η τελευταία εντολή μας οδηγεί σε μια σειρά προτροπών όπου μπορούμε να ελέγξουμε και να κάνουμε αλλαγές στις επιλογές ασφάλειας της εγκατάστασης της mysql. Για παράδειγμα μπορούμε να δοκιμάσουμε την ισχύ του κωδικού πρόσβασης για την βάση δεδομένων. Αφού ολοκληρώσουμε την εγκατάσταση σειρά έχει η δημιουργία ενός χρήστη και της βάσης δεδομένων για την εφαρμογή μας. Είναι καλή πρακτική για κάθε βάση δεδομένων που έχουμε να έχουμε και έναν χρήστη με τις απαραίτητες άδειες εκτέλεσης εντολών mysql για την αποφυγή προβλημάτων από κακόβουλους χρήστες.

Για να ολοκληρωθεί η εγκατάσταση έπρεπε να εισάγω τα δεδομένα της τοπικής βάσης δεδομένων σε αυτή του διακομιστή. Αρχικά έκανα εξαγωγή της τοπικής βάσης δεδομένων και κάνοντας χρήσης ftp(File Transfer Protocol) μετέφερα το αρχείο στο διακομιστή. Από εκεί έτρεξα την εντολή **mysql -u username -p database_name < file.sql** και κατά την ολοκλήρωση η εγκατάσταση και η εισαγωγή των δεδομένων της βάσης δεδομένων μου ήταν έτοιμη.

Ένωση βάσης δεδομένων με backend

Για πολλά χρόνια κατά τη δημιουργία εφαρμογών συνηθίζονταν να ενώνεται η βάση δεδομένων απευθείας με την εφαρμογή. Στις μέρες μας κάτι τέτοιο αποφεύγεται μιας και υπάρχουν πολλοί κίνδυνοι. Γενικά, για εφαρμογές με τεράστιες βάσεις δεδομένων γίνεται κατανοητό από πολύ νωρίς ότι οι απλές εφαρμογές που προσφέρει μια βάση δεδομένων (εισαγωγή, ενημέρωση, ανάκτηση) δεν είναι αρκετές. Επομένως πρέπει να δημιουργηθούν συγκεκριμένες διαδικασίες για την ενημέρωση περίπλοκων στοιχείων απευθείας στη βάση δεδομένων. Κάτι το οποίο δεν είναι πολύ καλό. Οι βάσεις δεδομένων προσφέρουν εξαιρετική απόδοση κατά την εκτέλεση εντολών εισαγωγής, ενημέρωσης, ανάκτησης και διαγραφής γνωστές και ως CRUD (Create, Read, Update, Delete) αλλά όχι για μακροχρόνιες διαδικασίες. Επίσης, αν στο μέλλον η μετάβαση των δεδομένων σε μια άλλη βάση δεδομένων, όπου η έννοια των μακροχρόνιων διαδικασιών δεν υποστηρίζεται, κάνει τη συνεχή βελτίωση του συστήματος πολύ δύσκολη. Έτσι η παροχή άμεσης σύνδεσης της βάσης δεδομένων με την εφαρμογή δεν είναι καλή ιδέα, μιας και όποιος έχει πρόσβαση σε αυτή την εφαρμογή μπορεί δυνητικά να αποκτήσει πρόσβαση στην βάση δεδομένων και να διαγράψει ή να μετατρέψει σημαντικά δεδομένα (14).

REST API



Πλέον συνηθίζεται να υπάρχει ένας μεσολαβητής ανάμεσα στην βάση δεδομένων και την εφαρμογή. Αυτή η διαδικασία ακολουθήθηκε και στην παρούσα εφαρμογή.

Τα πλεονεκτήματα αυτής της διαδικασίας είναι:

1. Η υλοποίηση της εφαρμογής για άλλες πλατφόρμες είναι πιο εύκολη (π.χ. κινητές συσκευές).
2. Η λογική της εφαρμογής βρίσκονται σε μια κεντρική βάση κώδικα και είναι πιο εύκολη η προσθήκη και αφαίρεση λειτουργιών ή ακόμη και η αλλαγή τύπου βάσης δεδομένων χωρίς ο χρήστης να χρειαστεί να εγκαταστήσει νέα έκδοση της εφαρμογής.
3. Είναι αυξημένη η ασφάλεια του συστήματος.
4. Είναι πιο εύκολη η συντήρηση της εφαρμογής (Αν υπάρχει κάποιο πρόβλημα στη λογική της εφαρμογής μπορεί να λυθεί πολύ απλά χωρίς οι χρήστες να χρειαστεί να εγκαταστήσουν άλλη έκδοση)

Τα μειονεκτήματα αυτής της διαδικασίας είναι:

1. Θα χρειαστεί περισσότερος χρόνος για την υλοποίηση της εφαρμογής.
2. Είναι πιο περίπλοκο για νέους προγραμματιστές να κατανοήσουν την ιδέα ενός μεσολαβητή.
3. Ο διακομιστής αναγκάζεται να λαμβάνει μεγαλύτερο φόρτο (Αν και στις μέρες μας με τις αποδόσεις που έχουν οι διακομιστές αυτό δεν αποτελεί ιδιαίτερο πρόβλημα. Αντιθέτως συνηθίζεται να αποφεύγεται η εκτέλεση περίπλοκων και ενεργοβόρων διαδικασιών από τη συσκευή του χρήστη. Ειδικά αν αυτή είναι κινητή συσκευή).

Για τη δημιουργία αυτού του μεσολαβητή χρησιμοποίησα NodeJS με typescript και express. Η NodeJS είναι ένα περιβάλλον διακομιστή ανοιχτού κώδικα το οποίο μπορεί να εγκατασταθεί σε διάφορες πλατφόρμες (Windows, Linux, macOS). Είναι γραμμένη σε python και ο προγραμματιστής κάνοντας χρήση της γλώσσας JavaScript μπορεί να υλοποιήσει εύκολα και γρήγορα τον δικό του διακομιστή. Η express είναι μια δομή η οποία συνεργάζεται με την NodeJS και προσφέρει εύκολη και γρήγορη υλοποίηση μεσολαβητή γνωστού και ως REST API (Representational state transfer - Application programming interface). Το API είναι μια διεπαφή από ένα σύνολο κανόνων που επιτρέπουν στις εφαρμογές να επικοινωνούν μεταξύ τους, ενώ REST πρόκειται για ένα σύνολο κανόνων που πρέπει να ακολουθεί ένας προγραμματιστής κατά τη δημιουργία του API.

Ένας από αυτούς τους κανόνες δηλώνει ότι μπορείτε να ανακτήσετε ένα συγκεκριμένο κομμάτι δεδομένων όταν κάποιος συνδέεται σε μια συγκεκριμένη διεύθυνση διαδικτύου.

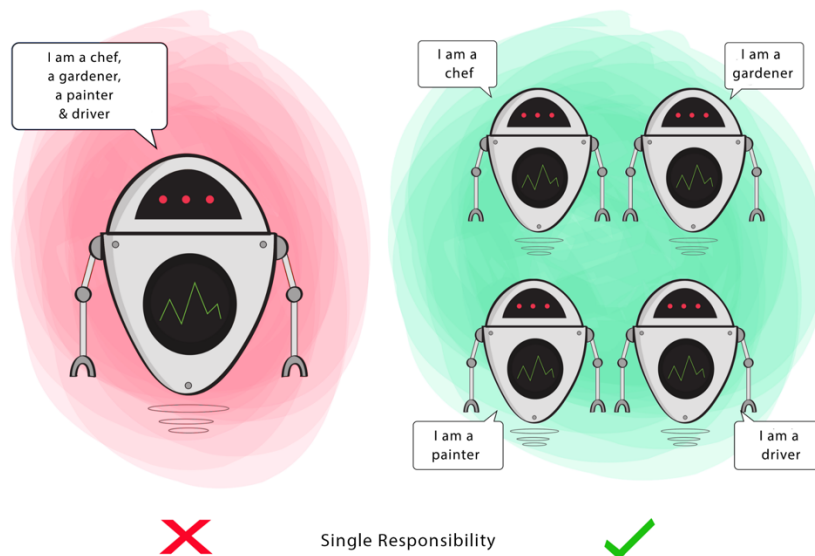
Η TypeScript είναι μια γλώσσα προγραμματισμού ανοιχτού κώδικα. Πρόκειται για μια επέκταση της JavaScript και προσθέτει λειτουργίες σε αυτή κάνοντας πιο εύκολη την περιγραφή ενός αντικειμένου επιβεβαιώνοντας ότι ακολουθείται ο αντικειμενοστραφής προγραμματισμός. Για να σιγουρευτώ ότι η εφαρμογή μου θα είναι εύκολα επεκτάσιμη και ότι θα αποφευχθούν λάθη στον κώδικα έπρεπε να ακολουθήσω συγκεκριμένα design patterns (σχεδιαστικά πρότυπα). Design pattern είναι τυποποιημένες βέλτιστες πρακτικές που κάποιος προγραμματιστής μπορεί να ακολουθήσει για την επίλυση συγκεκριμένων προβλημάτων που μπορεί να αντιμετωπίσει κατά την υλοποίηση της εφαρμογής του.

Για το API χρησιμοποίησα τα παρακάτω Design Patterns:

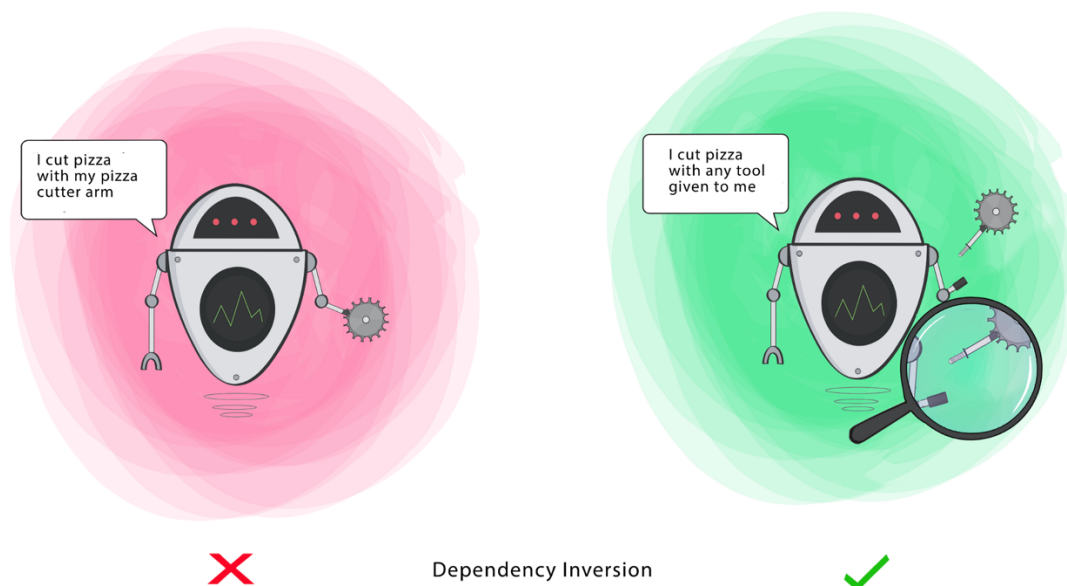
- Repository Pattern. Μιας και η βασική λειτουργία του API είναι η ένωση του με την βάση δεδομένων η χρήση του Repository Pattern ήταν αναγκαία. Πρόκειται για αποθετήρια ή κλάσεις όπου ενσωματώνουν την λογική που απαιτείται για την

επικοινωνία με την βάση δεδομένων. Συγκεντρώνουν την κοινή λειτουργικότητα για την πρόσβαση στα δεδομένα προσφέροντας καλύτερη συντήρηση και αποσπά την τεχνολογία που χρησιμοποιείται για την βάση δεδομένων από την λογική της εφαρμογής. Για παράδειγμα αν στο μέλλον αποφασίσω αντί για mysql να χρησιμοποιήσω oracle το μόνο που έχω να κάνω είναι να αλλάξω ένα αρχείο. Σε κάθε άλλη περίπτωση και αν δεν είχα κάνει χρήση αυτού του pattern θα έπρεπε να κάνω αμέτρητες αλλαγές για να ξανά γίνει το API λειτουργικό.

- Single Responsibility Pattern. Μια κλάση πρέπει να έχει ενιαία ευθύνη σχετικά με το τι κάνει. Εάν μια κλάση έχει πολλές ευθύνες, οι πιθανότητες για προβλήματα στον κώδικα αυξάνονται μιας και μια μικρή αλλαγή σε μια ευθύνη μπορεί να επηρεάσει τις άλλες χωρίς να το γνωρίζουμε.



- Open-Closed pattern. Η λογική αυτού του pattern είναι ότι οι κλάσεις πρέπει να είναι ανοιχτές για επέκταση αλλά κλειστές για τροποποίηση. Σε περίπτωση μετατροπής των λειτουργιών μιας κλάσης θα επηρεαστούν οι λειτουργίες όλων των άλλων κλάσεων που την χρησιμοποιούν και δυνητικά θα υπάρξουν προβλήματα που συνήθως είναι πολύ δύσκολο να λυθούν χωρίς ανακατασκευή μεγάλων σημείων του κώδικα.
- Dependency Inversion. Η κλάση που εκτελεί μια ενέργεια δεν πρέπει να εξαρτάται από το εργαλείο που απαιτείται για την εκτέλεση αυτής της ενέργειας. Θα πρέπει να υπάρξει μια ενδιάμεση διεπαφή η οποία θα ενώσει την κλάση με το εργαλείο. Είναι σημαντικό να σημειωθεί πως τόσο η κλάση όσο και η διεπαφή δεν πρέπει να «γνωρίζουν» πώς λειτουργεί το εργαλείο.



- **Model-Route -Service.** Προκειμένου ο κώδικας να είναι καθαρός και κάθε κλάση να ακολουθεί το single responsibility pattern ακολουθήσα το Model-Route -Service όπου κάθε ένα από αυτά έχει μια ξεχωριστή λειτουργία. Για παράδειγμα το model περιγράφει τα δεδομένα που υπάρχουν στην βάση δεδομένων μέσω μιας διεπαφής και ενώνεται με τα repositories στο repository pattern. Το service ενώνεται με τη διεπαφή του εκάστοτε model και με το repository για το συγκεκριμένο model και περιέχει όλη την λογική της εφαρμογής και τις λειτουργίες. Ενώ τέλος το route είναι αυτό που καθορίζει το ποια συνάρτηση θα τρέξει κάθε φορά που κάποιος χρήστης επισκέπτεται μια σελίδα. Το route δεν πρέπει να περιέχει τίποτα από την λογική της εφαρμογής, παρά μόνο να κατανέμει το φόρτο εργασίας στην εκάστοτε συνάρτηση και να περιμένει μια απάντηση από αυτή.

Πέραν από το private API το οποίο χρησιμοποιείται καθαρά για την επικοινωνία της βάσης δεδομένων με την εφαρμογή QBI δημιούργησα ένα public api όπου μπορεί ο οποιοσδήποτε έχει λογαριασμό στο QBI να έχει πρόσβαση στα δεδομένα με ασφάλεια κάνοντας χρήση του API key που του προσφέρεται δωρεάν από το QBI. Έτσι, οποιοσδήποτε έχει λογαριασμό QBI μπορεί να χρησιμοποιήσει ένα μοναδικό κλειδί και να αποκτήσει πρόσβαση σε συγκεκριμένες εντολές στο REST API. Με αυτόν τον τρόπο γίνεται εύκολο σε κάποιον που κατέχει προγραμματιστικές ικανότητες να δημιουργήσει μια εφαρμογή για κινητά ή ακόμη και δικό του web application όπου θα μπορεί να δείχνει τα κτίρια τα οποία σχετίζονται με τον λογαριασμό του ή ακόμη και άλλα. Για παράδειγμα μπορεί κάποιος να δημιουργήσει μια εφαρμογή για κινητά η οποία να αναγνωρίζει το περιεχόμενο ενός Qr Code και από αυτό θα εξάγει το αναγνωριστικό του κτιρίου και θα καλεί την απαραίτητη συνάρτηση στο public API. Με τα δεδομένα που θα του επιστραφούν μπορεί να δείξει στην εφαρμογή του με όμορφο τρόπο το περιεχόμενο κάποιου κτιρίου χωρίς να χρειαστεί να δημιουργήσει όλη τη δική του βάση δεδομένων.

Ανάλυση λειτουργιών Public Api

Υπάρχουν 4 διαφορετικές εντολές με τις οποίες κάποιος προγραμματιστής μπορεί να αντλήσει πληροφορίες μέσω του Public Api. Το μόνο που θα χρειαστεί για να το κάνει αυτό είναι η δημιουργία ενός API token μέσα από την εφαρμογή και με την χρήση αυτού στην

διεύθυνση <https://api.theqbi.com> μπορεί να στέλνει αιτήματα στην διεπαφή και να λαμβάνει απαντήσεις.

api/public/v1/sites/	JSON	The all the sites assigned to your user	'x-api-key':5tlDVo872k4c6bZoL0MdSqPGwkNIPxNW
----------------------	------	---	--

Με την πρώτη εντολή κάποιος χρήστης μπορεί να λάβει πληροφορίες για όλες τις εγγραφές οι οποίες έχουν δημιουργηθεί με τον δικό του λογαριασμό. Ένα παράδειγμα φαίνεται στην παρακάτω φωτογραφία.

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** https://api.theqbi.com/api/public/v1/sites/
- Headers:** x-api-key: 5tlDVo872k4c6bZoL0MdSqPGwkNIPxNW
- Status:** 201 Created
- Time:** 387 ms
- Size:** 23.05 KB

The response body is a JSON array of site objects. The first object is:

```
{
  "id": 105,
  "site_id": 16,
  "title": "Πληροφορίες",
  "content": "Το κτήριο Αρετάας φιλοξενεί δύο τμήματα του Ιονίου Πανεπιστημίου, το Τμήμα Πληροφορικής και το Τμήμα Τεχνών Ήχου και Εικόνας. Παρέχει στους φοιτητές αίθουσες διδασκαλίας, αλλά και 2 εργαστήρια για το Τμήμα Πληροφορικής και 2 αίθουσες διδασκαλίας για το Τμήμα Τεχνών Ήχου και Εικόνας. Επίσης, υπάρχει το γραφείο Υποστήριξης Συστημάτων του Τμήματος Πληροφορικής. Η ιδιαιτερότητα του κτηρίου βρίσκεται στο ισόγειο, όπου υπάρχει το ερευνητικό εργαστήριο Βιοπληροφορικής και Ανθρώπινης Ηλεκτροφυσιολογίας. Το εργαστήριο αυτό χρησιμοποιεί το Τμήμα Πληροφορικής για τις ερευνητικές του δραστηριότητες σχετικά με τις νευρολογικές διαταραχές και τη συσχέτισή τους με τις υποκτυπιακές μετρήσεις βιοενέργειας (πληροφορίες παρακάτω). Τα γραφεία των μελών ΔΕΠ των τμημάτων βρίσκονται σε διαφορετικό χώρο, στο Ανάκτορο τωvλnAy. Μιχαήλ και Γεωργίου, ενώ η βιβλιοθήκη των τμημάτων βρίσκεται στη Κεντρική Βιβλιοθήκη του Πανεπιστημίου."
}
```

api/public/v1/sites/location/:lat/:lng	JSON	Get near by sites (The default accuracy is 100km)	'x-api-key':5tlDVo872k4c6bZoL0MdSqPGwkNIPxNW
--	------	---	--

Με αυτή την εντολή κάποιος χρήστης παίρνει πληροφορία σχετικά με κτίρια τα οποία βρίσκονται σε ακτίνα 100 χιλιομέτρων από μια συγκεκριμένη τοποθεσία, όπως φαίνεται στην παρακάτω φωτογραφία.

GET https://api.theqbi.com/api/public/v1/sites/location/39.466084/20.971219

Send Save

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Code

Headers 7 hidden

KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
☒ x-api-key	5tIDVo872k4c6bZoL0MdSqPGwkNIPxNW				
Key	Value	Description			

Body Cookies Headers (9) Test Results Status: 200 OK Time: 477 ms Size: 28.43 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 21,
3   "userId": 11,
4   "credential": "Konδvn",
5   "latitude": 39.476270473142975,
6   "longitude": 20.84709995775484,
7   "createdAt": "2020-11-23T21:09:37.000Z",
8   "updatedAt": "2020-11-23T21:09:37.000Z",
9   "deletedAt": null,
10  "SiteImages": [
11    {
12      "id": 179,
13      "site_id": 21,
14      "alt": "dfc8b320dd5fd9716b32abd91e69ec71",
15      "file": "dfc8b320dd5fd9716b32abd91e69ec71",
16      "is_main": 0,
17      "createdAt": "2020-11-23T21:13:30.000Z",
18      "updatedAt": "2020-11-23T21:13:30.000Z",
19      "deletedAt": null
20    }
21  ]
22 }
```

api/public/v1/sites/location/:lat/:lng/:accuracy JSON Get sites near location with an accuracy 'x-api-key':5tIDVo872k4c6bZoL0MdSqPGwkNIPxNW

Η επόμενη εντολή είναι παρόμοια με την προηγούμενη, με τη μόνη διαφορά πως ο χρήστης μπορεί να λάβει πληροφορίες για κτίρια γύρω από μια ακτίνα που ο ίδιος έχει ορίσει σε χιλιόμετρα.

GET https://api.theqbi.com/api/public/v1/sites/location/39.466084/20.971219/1000

Send Save

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Code

Headers 7 hidden

KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
☒ x-api-key	5tIDVo872k4c6bZoL0MdSqPGwkNIPxNW				
Key	Value	Description			

Body Cookies Headers (9) Test Results Status: 200 OK Time: 431 ms Size: 28.43 KB Save Response

Pretty Raw Preview Visualize JSON

```
242 {
243   "id": 171,
244   "site_id": 19,
245   "alt": "f9ac70f7238e17486af5403bdda42e72",
246   "file": "f9ac70f7238e17486af5403bdda42e72",
247   "is_main": 0,
248   "createdAt": "2020-10-28T19:50:24.000Z",
249   "updatedAt": "2020-10-28T19:50:24.000Z",
250   "deletedAt": null
251 },
252 {
253   "id": 170,
254   "site_id": 19,
255   "alt": "ae8c4e93d3f763d7c77db71e96c4fb0d",
256   "file": "ae8c4e93d3f763d7c77db71e96c4fb0d",
257   "is_main": 0,
258   "createdAt": "2020-10-28T19:50:24.000Z",
259   "updatedAt": "2020-10-28T19:50:24.000Z",
260   "deletedAt": null
261 }
```

api/public/v1/sites/:id JSON Get one site with an id 'x-api-key':5tIDVo872k4c6bZoL0MdSqPGwkNIPxNW

Η τελευταία διαθέσιμη εντολή του Public Απi του QBI είναι αυτή που δίνει όλες τις πληροφορίες για ένα μόνο κτίριο. Έτσι, αν κάποιος για παράδειγμα δημιουργήσει μια εφαρμογή η οποία σαρώνει ένα απλό κώδικα Qr, μπορεί κάνοντας χρήση της παραπάνω

εντολής να λάβει τις πληροφορίες που θέλει σε μορφή json και να τις προβάλει στις εφαρμογή του όπως αυτός επιθυμεί.

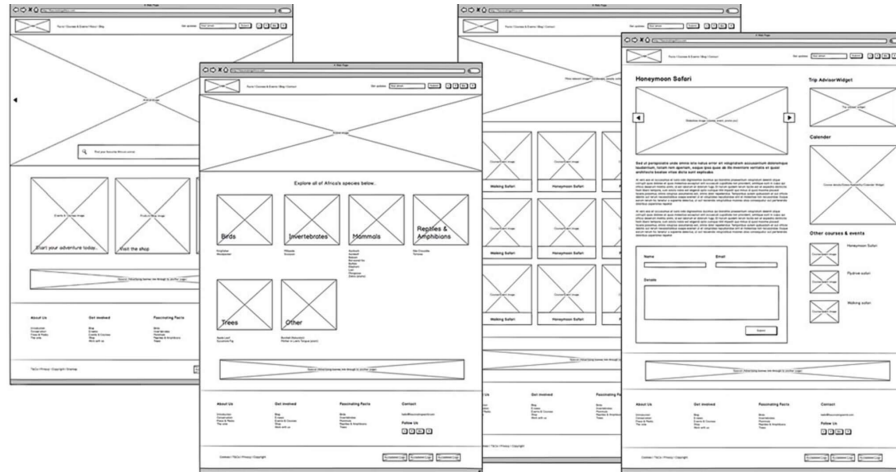
The screenshot shows a REST client interface with a GET request to `https://api.theqbl.com/api/public/v1/sites/21`. The `Headers` tab is active, showing a single header `x-api-key` with the value `5tdDVa872k4c6bZoLOMdSqPGwkNIPxNW`. The `Body` tab is also active, displaying a JSON response in the `Pretty` view. The response status is `200 OK` with a time of `282 ms` and a size of `5.71 KB`.

```
1 {
2   "id": 21,
3   "userId": 11,
4   "credential": "Komdvn",
5   "latitude": 39.476270473142975,
6   "longitude": 20.84709995775484,
7   "createdAt": "2020-11-23T21:09:37.000Z",
8   "updatedAt": "2020-11-23T21:09:37.000Z",
9   "deletedAt": null,
10  "SiteImages": [
11    {
12      "id": 177,
13      "site_id": 21,
14      "alt": "f462a1fc5f605200b3c34fb83322bea2",
15      "file": "f462a1fc5f605200b3c34fb83322bea2",
16      "is_main": 0,
17      "createdAt": "2020-11-23T21:13:30.000Z",
18      "updatedAt": "2020-11-23T21:13:30.000Z",
19      "deletedAt": null
20    },
21    {
22      "id": 178,
23      "site_id": 21,
```

Δημιουργία frontend

Η δημιουργία του frontend, δηλαδή της εφαρμογής που τελικά βλέπει ο χρήστης, είναι μια διαδικασία που πρέπει να περάσει από αρκετά στάδια ώστε να μπορέσει κάποιος να περιηγηθεί σε αυτή και να έχει μια ευχάριστη εμπειρία.

Αρχικά, έπρεπε να δημιουργηθούν σχέδια για το πώς θα ήθελα να φαίνεται η εφαρμογή. Τέτοια σχέδια είναι και τα wireframes, απλά τετράγωνα και γραμμές τα οποία τελικά δίνουν την βασική μορφοποίηση της εφαρμογής.



Στο επόμενο στάδιο έπρεπε να επιλέξω το εργαλείο με το οποίο θα υλοποιήσω την εφαρμογή. Στην συγκεκριμένη περίπτωση επέλεξα την Angular ένα framework ανοιχτού κώδικα, διότι προσφέρει εύκολη δημιουργία σελίδων έχοντας ταυτόχρονα πολλά έτοιμα μέρη τα οποία με μικρές τροποποιήσεις μπορείς να τα εντάξεις στην εφαρμογή σου. Βέβαια ένα από τα μεγαλύτερα πλεονεκτήματα της είναι το ότι πρόκειται για ένα έργο το οποίο υποστηρίζεται από την Google. Αυτό διασφαλίζει το γεγονός ότι θα συνεχίσει να υπάρχει και να εξελίσσεται και πως δεν θα χρειαστεί στο μέλλον να ξανά γράψεις όλη την εφαρμογή σου από την αρχή με κάποιο άλλο εργαλείο παρά μόνο θα το συντηρείς και θα το αναπτύσσεις (1).

Η Angular οργανώνει τον κώδικα σε ενότητες, οι οποίες διευκολύνουν την οργάνωση των λειτουργιών της εφαρμογής, διαχωρίζοντάς την σε χαρακτηριστικά και επαναχρησιμοποιήσιμα κομμάτια κώδικα.

Η υλοποίηση της εφαρμογής από άποψη κώδικα έχει δημιουργηθεί έχοντας στο μυαλό το architectural pattern MVVM (Model, View, View-Model) και το Observer Pattern.

Το MVVM χωρίζει την εφαρμογή σε τρεις βασικές υποκατηγορίες. Τα δεδομένα και την πληροφορία (Model), το περιεχόμενο που θα είναι ορατό στο χρήστη (View), και το ενδιάμεσο κομμάτι που ενώνει το Model και το View και περιέχει την λογική της εφαρμογής (View-Model) (2).

Model

Το model είναι κατά βάση ένα αντικείμενο το οποίο αντιπροσωπεύει τα πραγματικά δεδομένα ή τις πληροφορίες που θέλουμε να διαχειριστούμε μέσα στην εφαρμογή. Ένα παράδειγμα ενός Model μπορεί να είναι μια επαφή στο κινητό μας τηλέφωνο, η οποία περιέχει όνομα, αριθμό τηλεφώνου, διεύθυνση κλπ. Το σημαντικό για ένα Model είναι να περιέχει μόνο πληροφορίες και καθόλου λογική της εφαρμογής ή κάποια υπηρεσία. Είναι η μεγαλύτερη πρόκληση που μπορεί να αντιμετωπίσει κάποιος κατά την υλοποίηση μιας εφαρμογής το να κρατήσει ένα Model “καθαρό» (2).

View

Το View είναι αυτό με το οποίο τελικά γίνεται η αλληλεπίδραση του χρήστη με την εφαρμογή μας. Πρόκειται για την παρουσίαση των δεδομένων που έρχονται από το Model και το πώς αυτά θα εμφανιστούν τελικά στο χρήστη. Για παράδειγμα μια ημερομηνία στο Model μπορεί να περιέχει μόνο αριθμούς και όχι κάποια εύκολα αναγνωρίσιμη πληροφορία από τον χρήστη, το View είναι αυτό που θα κάνει αυτή την πληροφορία αναγνώσιμη. Το View μπορεί να περιέχει και τον απαραίτητο κώδικα ώστε ο χρήστης να αλληλοεπιδρά με την εφαρμογή μας και τελικά να ανανεώνεται το Model, όπως για παράδειγμα κουμπιά και φόρμες (2).

View-Model

Πρόκειται για το βασικό κομμάτι, διότι εισάγει την έννοια της αποκοπής του View από το Model. Αντί να ενημερώνεται το View από το Model για αλλαγές σε δεδομένα την διαδικασία αυτή την αναλαμβάνει το View-Model. Το View μπορεί να λάβει κάποια δεδομένα από τον χρήστη, έπειτα το View-Model θα τα επεξεργαστεί και θα μετατρέψει τα δεδομένα σε κάτι το οποίο το Model γνωρίζει και μπορεί να αποθηκεύσει.

Ένα άλλο pattern που χρησιμοποίησα είναι το Observer pattern. Κάτι το οποίο συναντάται πολύ συχνά σε εφαρμογές που έχουν χτιστεί με MVVM. Η κύρια λειτουργία του είναι πως ένα αντικείμενο διατηρεί μια λίστα από εξαρτώμενα αντικείμενα, που ονομάζονται παρατηρητές, και τους ειδοποιεί αυτόματα για τυχόν αλλαγές που θα προκύψουν σε κάποιο αντικείμενο.

Αφού τελείωσα την υλοποίηση της εφαρμογής τοπικά, και βεβαιώθηκα πως όλες οι λειτουργίες δουλεύουν σωστά έπρεπε να μεταφέρω την εφαρμογή μου σε κάποιον διακομιστή. Όπως ανέφερα στο κεφάλαιο που εξήγησα πώς έγινε η εγκατάσταση της βάσης δεδομένων στο διακομιστή, χρειάστηκε να κάνω κάποια βήματα για να είναι η εφαρμογή μου θεατή από το κοινό. Αρχικά, έπρεπε να εγκαταστήσω στο διακομιστή git ώστε να μπορώ κάθε φορά να ανανεώνω εύκολα την εφαρμογή αφού έχω πρώτα υλοποιήσει ή διορθώσει κάτι τοπικά από τον υπολογιστή μου. Αφού είχα τον κώδικα της εφαρμογής μου στο διακομιστή έτοιμο να χτιστεί έπρεπε να εγκαταστήσω την Angular και να τρέξω την εντολή **ng build –prod**. Με αυτή την εντολή η Angular δημιουργούσε τα απαραίτητα αρχεία και έκανε βελτιστοποιήσεις για να μπορέσει η εφαρμογή μου να τρέχει γρήγορα και με ασφάλεια στο διακομιστή (2).

Προγραμματισμός προχωρημένων Λειτουργιών QBI

Πέραν από τις βασικές λειτουργίες της εφαρμογής όπως απλή δημιουργία χρήστη, προβολή ήδη υπάρχοντος υλικού και του public API υπάρχουν κάποιες λειτουργίες οι οποίες αξίζει να αναφερθούν ξεχωριστά. Μια από αυτές είναι η δημιουργία και η σύνδεση κάποιου χρήστη με τη λειτουργία SSO (Single Sign On). (3) Πρόκειται για μια λειτουργία που πάρα πολλές εφαρμογές προσφέρουν στους χρήστες της, και δίνει την δυνατότητα σε αυτούς να δημιουργήσουν ή να συνδεθούν στο λογαριασμό τους σε μια εφαρμογή χρησιμοποιώντας κάποιον ήδη υπάρχον λογαριασμό σε κάποιον πάροχο ελέγχου ταυτότητας. Για παράδειγμα μπορεί κάποιος χρήστης να δημιουργήσει λογαριασμό QBI μέσω του Google ή του Facebook λογαριασμού και να αποκτήσει πρόσβαση σε όλες τις λειτουργίες της εφαρμογής. Ένας από τους τύπους SSO είναι το (4) OAuth2 το οποίο χρησιμοποιήθηκε για την υλοποίηση αυτής της λειτουργίας του QBI. Είναι μια προδιαγραφή στην οποία πρέπει να βασίζεται κάποιος πάροχος ελέγχου ταυτότητας προκειμένου να μπορεί εύκολα και με ασφάλεια να δώσει πρόσβαση σε άλλες εφαρμογές ακολουθώντας συγκεκριμένους κανόνες (2).

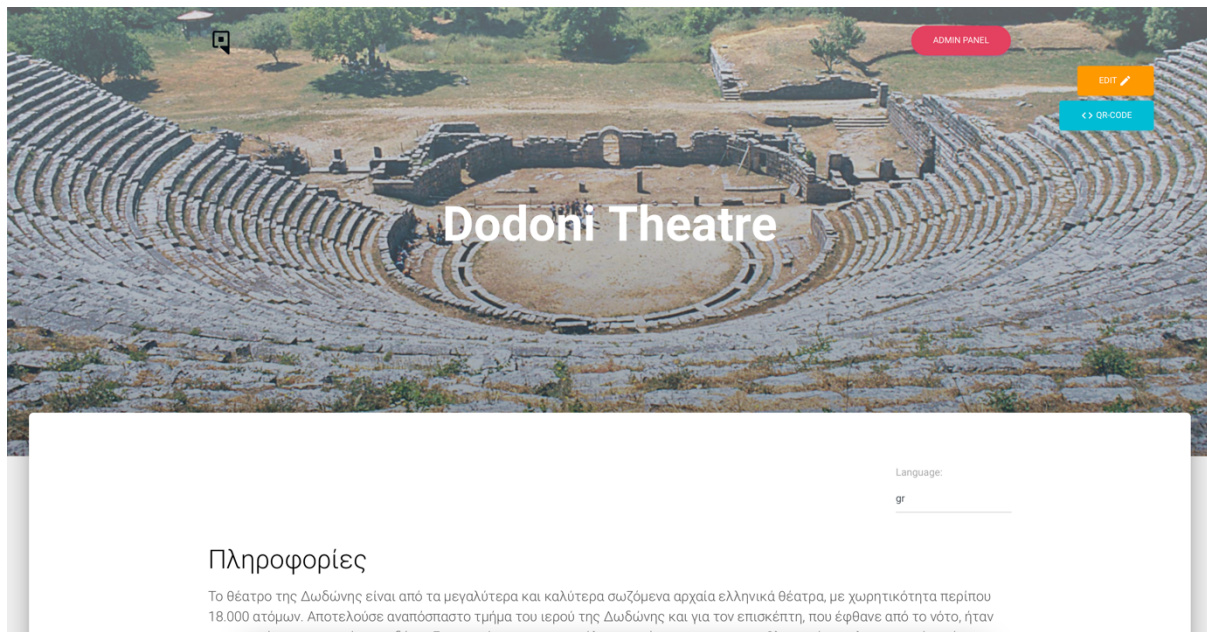
Τα πλεονεκτήματα του OAuth2 είναι (5):

- **Ευκολία** (Χρειάζεται να γνωρίζει κάποιος μονάχα έναν κωδικό και να αποκτήσει ασφαλή πρόσβαση σε όποια άλλη εφαρμογή υποστηρίζει αυτή τη λειτουργία)
- **Δικτύωση** (Λόγω της ταχύτητας που προσφέρει στη δημιουργία χρήστη προσελκύει περισσότερους επισκέπτες οι οποίοι τελικά κάνουν λογαριασμό στην εφαρμογή)
- **Ασφάλεια** (Όλες οι εφαρμογές πρέπει να ακολουθούν κάποιους κανόνες για να μπορέσουν να έχουν πρόσβαση στα δεδομένα του χρήστη)
- **Έλεγχος** (Των δεδομένων που μπορεί κάποια εφαρμογή να αποκτήσει για τον λογαριασμό)

Τα μειονεκτήματα του OAuth2 είναι:

- **Phishing** (Λόγω του ότι από ένα σημείο και μετά οι χρήστες εξοικειώνονται με την εύκολη σύνδεση σε άλλες εφαρμογές, μπορεί κάποιος να δημιουργήσει μια σελίδα η οποία να αποσκοπεί στην υποκλοπή των στοιχείων του χρήστη και τελικά πρόσβαση στο λογαριασμό του.)
- **Κατάχρηση δεδομένων** (Κακόβουλες εφαρμογές μπορεί να εκμεταλλευτούν την αθωότητα του χρήστη και να χρησιμοποιήσουν τα στοιχεία που τους δίνει ο πάροχος ελέγχου ταυτότητας με τρόπο τον οποίο δεν τον εγκρίνει ο χρήστης. Για παράδειγμα την πώληση του email) (5).

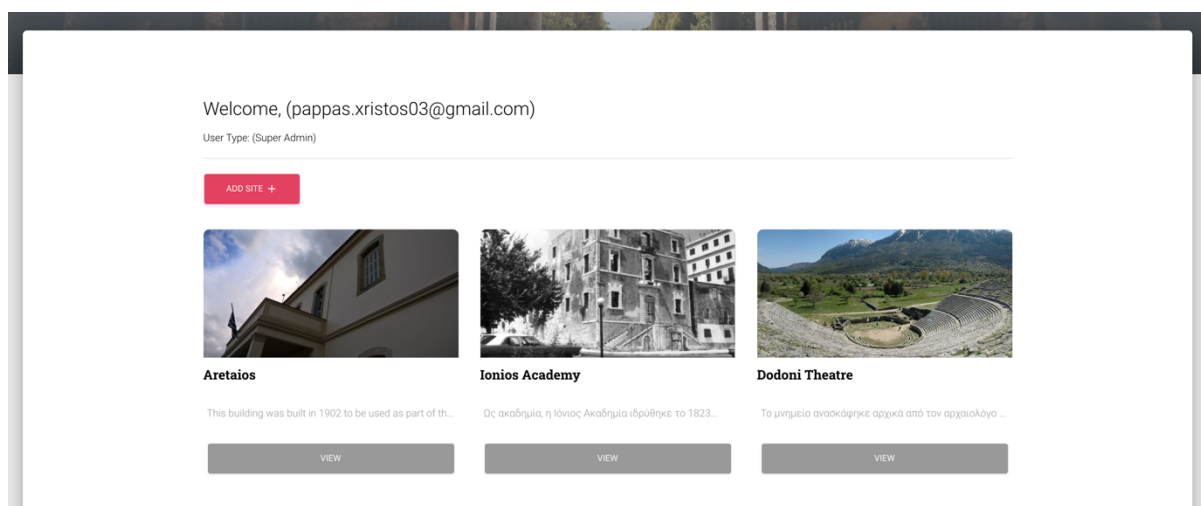
Μια άλλη λειτουργία του QBI είναι και η δημιουργία μοναδικού Qr Code για κάθε κτίριο. Κάνοντας χρήση μιας βιβλιοθήκης της Angular <https://www.npmjs.com/package/angularx-qrcode> και χρησιμοποιώντας τις συναρτήσεις της είναι εφικτό για κάποιον χρήστη να δημιουργήσει ένα μοναδικό Qr Code για το κτίριο του προκειμένου να βοηθήσει τους χρήστες να έχουν εύκολη πρόσβαση στο τι περιέχει κάποιο κτίριο ή τι πρεσβεύει ιστορικά και πολιτισμικά.





Δημιουργία εγκατάστασης πολιτισμικής κληρονομιάς/σημασίας από το διαχειριστικό

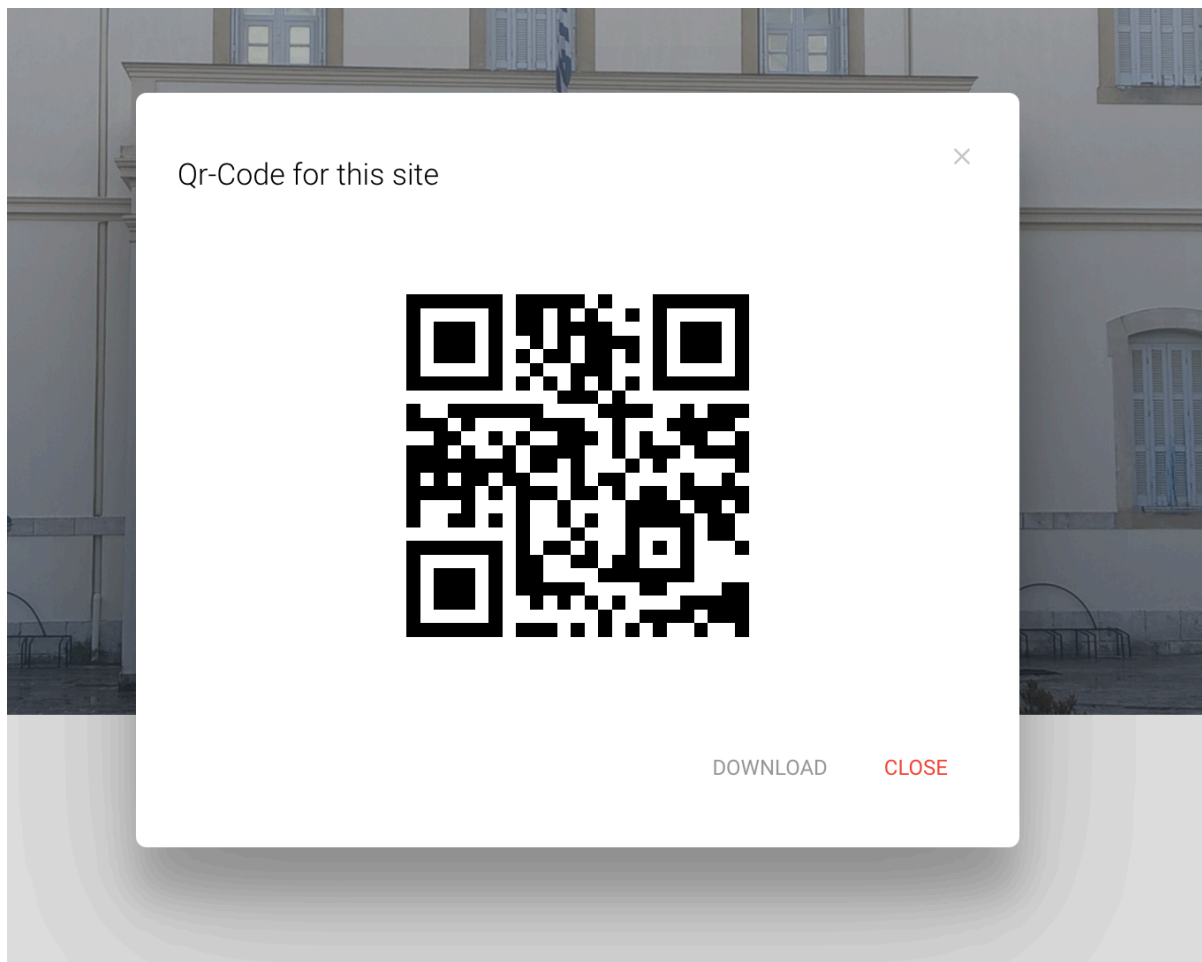
Κάθε χρήστης του QBI έχει πρόσβαση στο προσωπικό του διαχειριστικό. Μια σελίδα στην οποία μπορεί να δημιουργήσει καινούριες εγκαταστάσεις ή να επεξεργαστεί τις ήδη υπάρχουσες. Επιπλέον, μέσω αυτού του διαχειριστικού μπορεί να δημιουργήσει Token για να μπορέσει να κάνει χρήση του Public Api.



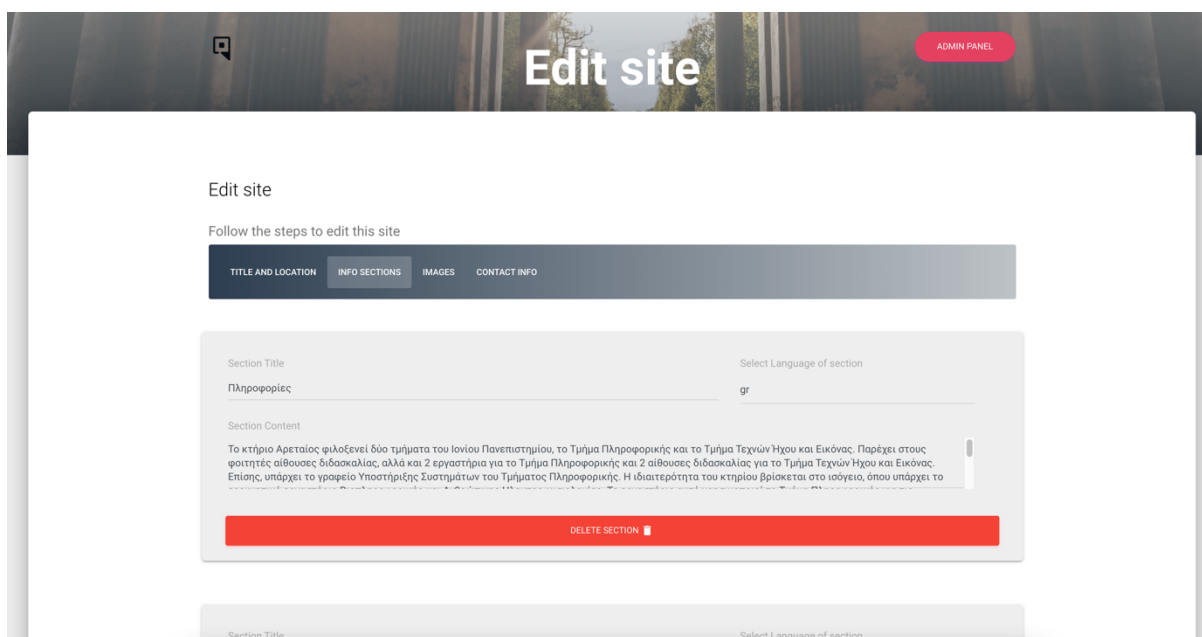
Όπως φαίνεται στην παραπάνω φωτογραφία ο συγκεκριμένος χρήστης έχει ήδη δημιουργήσει τρεις εγκαταστάσεις. Από αυτή τη σελίδα πατώντας την επιλογή “Add Site” μπορεί να δημιουργήσει καινούρια εγκατάσταση και πατώντας στην εκάστοτε επιλογή “View” μπορεί να δει, επεξεργαστεί ή να κατεβάσει το Qr Code που δημιούργησε η εφαρμογή για την συγκεκριμένη σελίδα.



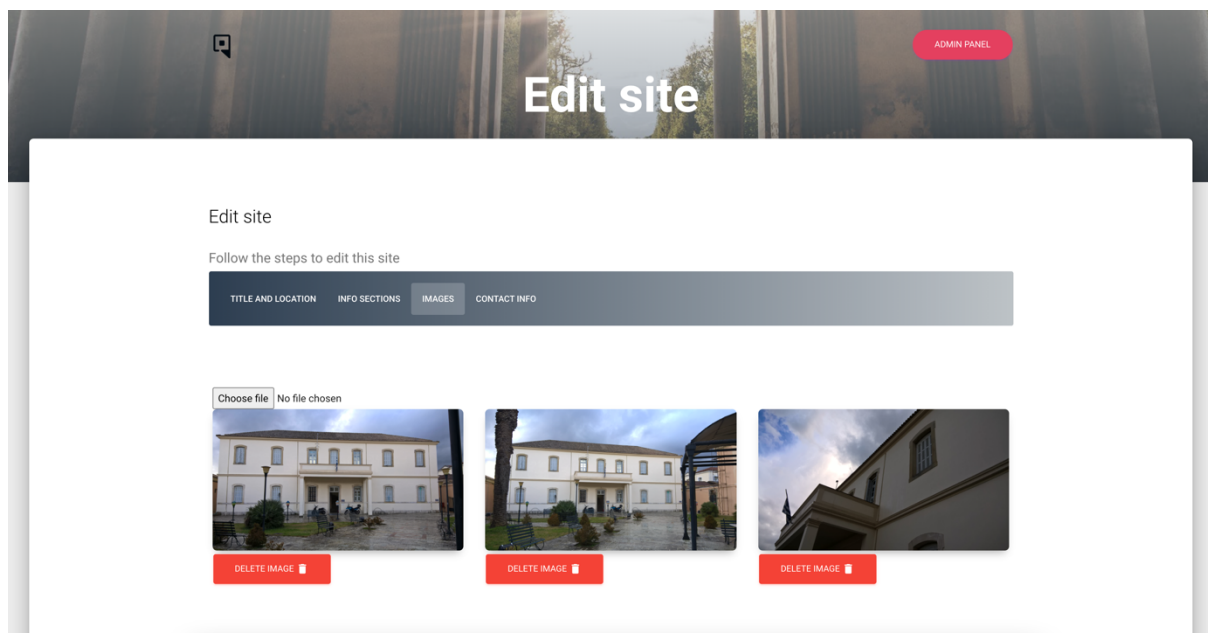
Πατώντας στην επιλογή “Edit” κάποιος χρήστης μπορεί να επεξεργαστεί την συγκεκριμένη σελίδα, ενώ πατώντας στην επιλογή “QR-CODE” μπορεί να δει και να κατεβάσει τον κώδικα Qr για την συγκεκριμένη σελίδα και να τον χρησιμοποιήσει όπως θέλει. Στις παρακάτω φωτογραφίες απεικονίζεται αναλυτικά οι προαναφερθείσες διαδικασίες.



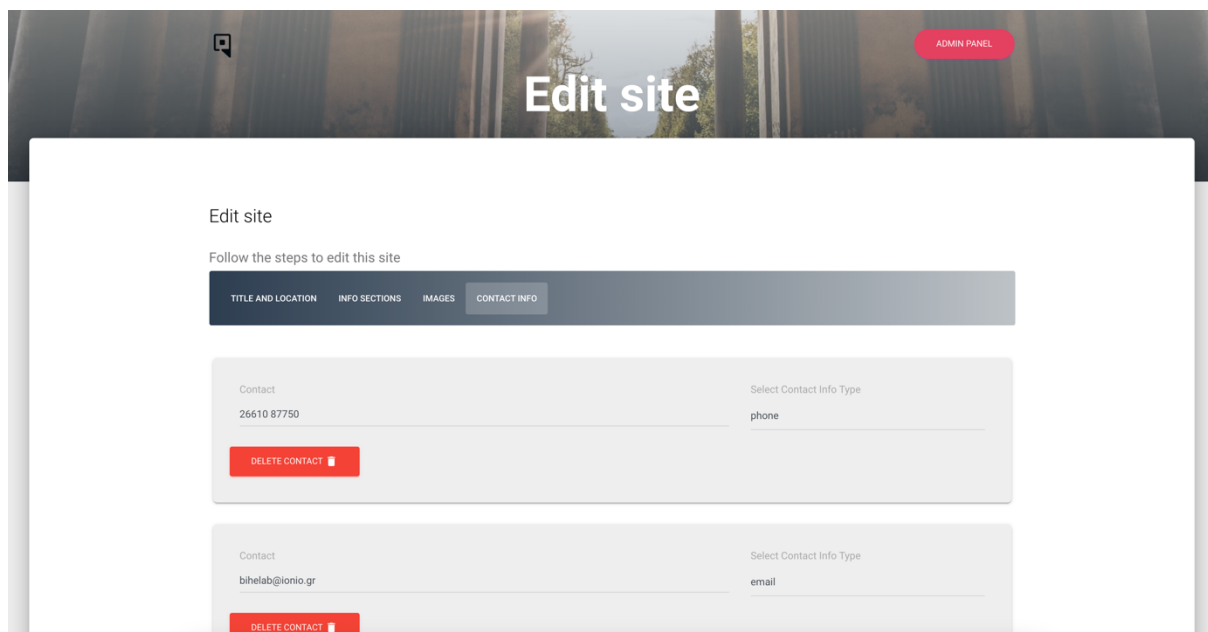
Προβολή και αποθήκευση μοναδικού Qr Code



Επεξεργασία, διαγραφή και αποθήκευση ενοτήτων και πληροφοριών



Επεξεργασία, διαγραφή, και ανέβασμα νέων φωτογραφιών



Επεξεργασία, διαγραφή και δημιουργία πληροφοριών επικοινωνίας,

Εγκατάσταση διακομιστή και δημιουργία ονόματος

Για τον διακομιστή επέλεξα να χρησιμοποιήσω Apache. Είναι ένας από τους πιο διαδεδομένους διακομιστές στον κόσμο. Παρέχει πολλές και ισχυρές δυνατότητες, όπως δυναμικά φορτώσιμες μονάδες, υποστήριξη πολυμέσων και υποστηρίζεται από πολλά λειτουργικά συστήματα. Η εγκατάστασή του σε ένα μηχάνημα linux είναι πολύ απλή και αρκούν μερικές εντολές στο τερματικό. Αρχικά πρέπει να ανανεώσουμε τα πακέτα που είναι αποθηκευμένα στο λειτουργικό σύστημα μέσω της εντολής `sudo apt update` και έπειτα με την εντολή `sudo apt install apache2` εγκαθιστούμε το λογισμικό στο λειτουργικό έτοιμο να το χρησιμοποιήσουμε(15).

Το επόμενο σημαντικό βήμα είναι η ρύθμιση του τείχους προστασίας προκειμένου να επιτρέψουμε την εξωτερική πρόσβαση σε συγκεκριμένες θύρες του διακομιστή. Η διαδικασία αυτή έγινε μέσω του **UFW** όπου είναι το προεπιλεγμένο εργαλείο διαμόρφωσης του τείχους προστασίας για το Linux. Έτσι με την εντολή `sudo ufw allow 'Apache'` αφήνουμε το λογισμικό να περάσει από το τείχος προστασίας. Μπορούμε να δούμε ότι η εντολή λειτούργησε γράφοντας `sudo ufw status` και θα πρέπει να μας επιστρέψει πίσω κάτι παρόμοιο με την παρακάτω φωτογραφία.

Output

Status: active

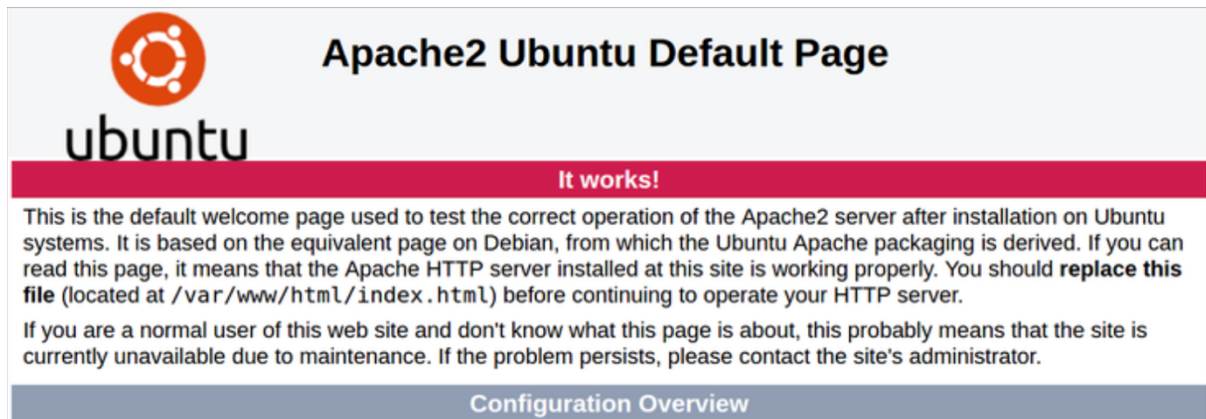
To	Action	From
--	-----	----
OpenSSH	ALLOW	Anywhere
Apache	ALLOW	Anywhere
OpenSSH (v6)	ALLOW	Anywhere (v6)
Apache (v6)	ALLOW	Anywhere (v6)

Τέλος πρέπει να δοκιμάσουμε ότι ο διακομιστής μας λειτουργεί και είναι έτοιμος να δεχτεί κάποια ιστοσελίδα. Για αυτό γράφουμε `sudo systemctl status apache2` και το αποτέλεσμα στο τερματικό μας θα πρέπει να έχει την παρακάτω μορφή της φωτογραφίας.

Output

```
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Drop-In: /lib/systemd/system/apache2.service.d
            └─apache2-systemd.conf
   Active: active (running) since Tue 2018-04-24 20:14:39 UTC; 9min ago
   Main PID: 2583 (apache2)
     Tasks: 55 (limit: 1153)
   CGroup: /system.slice/apache2.service
           └─2583 /usr/sbin/apache2 -k start
             └─2585 /usr/sbin/apache2 -k start
               └─2586 /usr/sbin/apache2 -k start
```

Έπειτα, απλά γράφουμε την διεύθυνση IP του διακομιστή μας σε κάποιον browser (φυλλομετρητή) και θα δούμε την παρακάτω εικόνα με την προκαθορισμένη κωδικοποίηση html.



Εφόσον η εγκατάσταση και ρύθμιση του Apache έχει τελειώσει πρέπει να δημιουργηθούν τα domains και sub domains ώστε να καθίσταται πιο εύκολο σε κάποιον χρήστη να επισκεφθεί την ιστοσελίδα. Το βασικό domain της εφαρμογής είναι το <https://theqbi.com/>. Έπειτα από την αγορά του έπρεπε να κάνω κάποιες επιπλέον ρυθμίσεις στο διακομιστή. Επειδή πέραν της βασικής εφαρμογής έχει υλοποιηθεί και το API ήταν αναγκαία η δημιουργία Virtual Hosts (Εικονικός Κεντρικός Υπολογιστής) ώστε να χωρίζονται οι λειτουργίες των δύο αυτών ξεχωριστών οντοτήτων. Τα περιεχόμενα μιας ιστοσελίδας σε έναν διακομιστή Apache βρίσκονται συνήθως στο φάκελο /var/www/public_html, για να διαχωρίσω το API με την εφαρμογή χρειάστηκε να αλλάξω την δομή αυτή και να δημιουργήσω διαφορετικούς φακέλους. Για παράδειγμα, η εφαρμογή βρίσκεται στο φάκελο /var/www/theqbi.com/public_html/ ενώ το API στο φάκελο /var/www/api.theqbi.com/public_html/ για να επιτευχθεί αυτό αρκούσε να τρέξω τις δύο παρακάτω εντολές:

```
$ sudo mkdir -p /var/www/theqbi.com/public_html
$ sudo mkdir -p /var/www/api.theqbi.com/public_html
```

Μετά τη δημιουργία των φακέλων έπρεπε να ρυθμίσω τις άδειες αυτών των φακέλων ώστε να μπορώ να κάνω αλλαγές σε αυτούς τρέχοντας την εντολή `sudo chown -R $USER:$USER /var/www/qbi.com/public_html` για τον εκάστοτε φάκελο και `sudo chmod -R 755 /var/www` προκειμένου να διασφαλίσω ότι επιτρέπεται η πρόσβαση ανάγνωσης στο γενικό φάκελο του ιστού και σε όλα τα αρχεία και τους φακέλους που περιέχει, ώστε οι σελίδες τις εφαρμογής να μπορούν να προβάλλονται σωστά (18).

Για να μπορέσουν οι Virtual Hosts να λειτουργήσουν έπρεπε να δημιουργήσω κάποια αρχεία. Αυτά τα αρχεία καθορίζουν την πραγματική διαμόρφωση των Virtual Hosts και καθορίζουν τον τρόπο με τον οποίο θα ανταποκριθεί ο διακομιστής Apache σε διάφορα αιτήματα. Ο Apache έχει ένα προεπιλεγμένο αρχείο Virtual Host κατά την εγκατάστασή του το οποίο χρησιμοποίησα ως πρότυπο. Έτσι, αντέγραψα το συγκεκριμένο αρχείο που ονομάζεται 000-default.conf και δημιούργησα ένα πανομοιότυπο με το όνομα του πρώτου

Virtual Host που ήθελα να δημιουργήσω για την εφαρμογή χρησιμοποιώντας την εντολή `sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/theqbi.com.conf`. Έπειτα, επεξεργάστηκα το περιεχόμενο αυτού του αρχείου προκειμένου να ρυθμίσω τον διακομιστή κάθε φορά που θα λαμβάνει κάποιο αίτημα από <https://theqbi.com/> να μεταφέρει αυτό το αίτημα στο φάκελο όπου βρίσκεται η εφαρμογή όπως φαίνεται στην παρακάτω φωτογραφία.

```
<VirtualHost *:80>
    ServerAdmin admin@theqbi.com
    ServerName theqbi.com
    ServerAlias www.theqbi.com
    DocumentRoot /var/www/theqbi.com/public_html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Τα ίδια βήματα ακολούθησα και για τη δημιουργία Virtual Host για το API.

Αφού δημιούργησα τα απαραίτητα αρχεία, έπρεπε να ενεργοποιήσω τους Virtual Hosts ώστε να μπορέσει ο διακομιστής να δέχεται αιτήματα και να τα ανακατευθύνει στις σωστές μονάδες αλλά και να απενεργοποιήσω τον προ εγκατεστημένο Virtual Host του Apache. Για να το κάνω χρησιμοποίησα τις παρακάτω εντολές για την ενεργοποίηση.

```
$ sudo a2ensite theqbi.com.conf
$ sudo a2ensite api.theqbi.com.conf
```

Ενώ για την απενεργοποίηση χρησιμοποίησα την εντολή.

```
$ sudo a2dissite 000-default.conf
```

Προκειμένου όμως οι αλλαγές που έκανα να έχουν κάποιο εμφανές αποτέλεσμα έπρεπε να κάνω επανεκκίνηση της υπηρεσίας Apache με τις παρακάτω εντολές.

```
$ sudo systemctl restart apache2
$ sudo systemctl status apache2
```

Τέλος, χρειάστηκε να αλλάξω τις διευθύνσεις DNS (Domain Name System) ώστε να μπορεί κάποιος να επισκεφτεί την ιστοσελίδα μου δίνοντας σε ένα πρόγραμμα περιήγησης το όνομά της. DNS, είναι στην ουσία ο τηλεφωνικός κατάλογος του διαδικτύου. Ένας άνθρωπος έχει πρόσβαση σε πληροφορίες διαδικτυακά μέσω των domain names όπως το theqbi.com και οι DNS μεταφράζουν ένα domain name σε μια διεύθυνση IP ώστε τα προγράμματα περιήγησης να μπορέσουν να φορτώσουν τους πόρους του εκάστοτε διακομιστή.

Προκειμένου η εφαρμογή και το API να ανέβουν στο διακομιστή έπρεπε να μεταφέρω εκεί όλα τα αρχεία από τον υπολογιστή μου. Για να γίνει αυτό με ασφάλεια και να βεβαιωθώ πως

τίποτα δεν θα πάει στραβά για παράδειγμα κατά τη διάρκεια μεταφόρτωσης όλων των αρχείων του υπολογιστή μου στο διακομιστή μέσω FTP χρησιμοποίησα git. Όπως ανέφερα σε προηγούμενο κεφάλαιο όπου ανέλυα τη διαδικασία δημιουργίας της εφαρμογής, χρησιμοποίησα το git ώστε να λαμβάνω τα αρχεία του κώδικα κάθε φορά που κάνω κάποια αλλαγή και έπειτα απλά ξανά δημιουργούσα τα απαραίτητα εκτελέσιμα αρχεία. Για να μπορέσει ο διακομιστής μου να επικοινωνήσει με τον διακομιστή git στο bitbucket έπρεπε να δημιουργήσω ένα κλειδί ssh και να το εγκαταστήσω στο διακομιστή git. Η δημιουργία ενός τέτοιου κλειδιού γίνεται με την εντολή **ssh-keygen** ακολουθώντας μερικά απλά βήματα στο τερματικό. Έπειτα βρίσκουμε τα αρχεία που περιέχουν το κλειδί (συνήθως στο φάκελο ~/.ssh/ για μηχανήματα Linux) και αντιγράφουμε το κλειδί `cat ~/.ssh/id_rsa.pub` ώστε να το εγκαταστήσουμε στο bitbucket.

Συμπεράσματα

Εισαγωγή

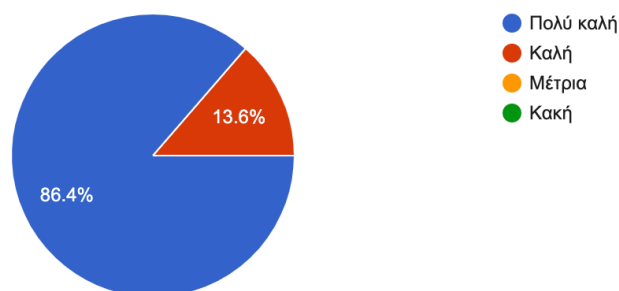
Για την αξιολόγηση της εφαρμογής μου δημιούργησα ένα ερωτηματολόγιο και το διένειμα σε 22 άτομα λαμβάνοντας τις απαντήσεις τους. Για την επιλογή των ατόμων βασίστηκα στην ηλικία τους και στην εξοικείωσή τους με την τεχνολογία. Επέλεξα κάποια από αυτά τα άτομα να είναι μηχανικοί λογισμικού και QA (Quality Assurance Engineers) ώστε να μπορέσω να συγκρίνω τις απαντήσεις τους με αυτές των ανθρώπων που ο επαγγελματικός τους κλάδος δεν συνάδει με την δημιουργία λογισμικού. Οι ερωτήσεις που επέλεξα είχαν να κάνουν με την σχεδίαση της εφαρμογής την ταχύτητα του διακομιστή, την προβολή της εφαρμογής σε κινητά μέσω της χρήσης Qr Code.

Αναλυτικά τα αποτελέσματα

Στην ερώτηση σχετικά με το πώς φάνηκε η εφαρμογή σχεδιαστικά το 86.4% απάντησε πως η σχεδίαση της εφαρμογής είναι **Πολύ Καλή** ενώ το 13.6% βρήκε την εφαρμογή σχεδιαστικά απλά **Καλή**. Τα άτομα τα οποία η απάντησή τους είναι **Καλή** έχουν ηλικία 41-55 ετών ενώ η εξοικείωσή τους με την τεχνολογία είναι αρκετά καλή.

Πως σας φάνηκε η εφαρμογή σχεδιαστικά;

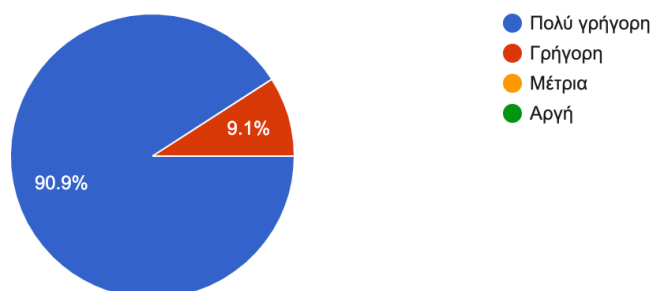
22 responses



Στην ερώτηση σχετικά με τη φόρτωση του περιεχομένου της σελίδας το 90.9% απάντησε ότι ήταν **Πολύ Γρήγορη** ενώ το 9.1% απάντησε πως ήταν απλά **Γρήγορη**. Σε αυτήν την περίπτωση παρατήρησα πως η εξοικείωση με την τεχνολογία (των ατόμων που απάντησαν πως η φόρτωση του περιεχομένου είναι Γρήγορη) είναι μέτρια. Από αυτό μπορώ να συμπεράνω πως ίσως άτομα τα οποία χρησιμοποιούν το διαδίκτυο σε συγκεκριμένες σελίδες κολοσσών όπως Google ή Facebook έχουν συνηθίσει το περιεχόμενο των σελίδων που επισκέπτονται να φορτώνει εξαιρετικά γρήγορα σε αντίθεση με αυτούς που η εξοικείωσή τους με την τεχνολογία είναι εξαιρετικά καλή και επισκέπτονται και άλλες, μη δημοφιλείς ιστοσελίδες.

Η φόρτωση του περιεχομένου της σελίδας ήταν:

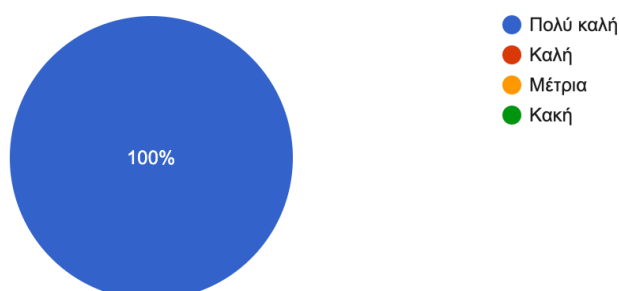
22 responses



Στην επόμενη ερώτηση ζήτησα από τα άτομα που έλαβαν μέρος στο ερωτηματολόγιο να σαρώσουν ένα Qr Code του οποίου το περιεχόμενο είχε να κάνει με το Αρχαίο Θέατρο της Δωδώνης και να κρίνουν την εφαρμογή όσον αναφορά την περιήγηση σε αυτή μέσω μιας κινητής συσκευής. Το 100% απάντησε πως ήταν **Πολύ Καλή**.

Σκανάρετε το παρακάτω QrCode με το κινητό σας (Dodoni Theatre). Η εμπειρία όσο αναφορά την όψη σε κινητή συσκευή ήταν:

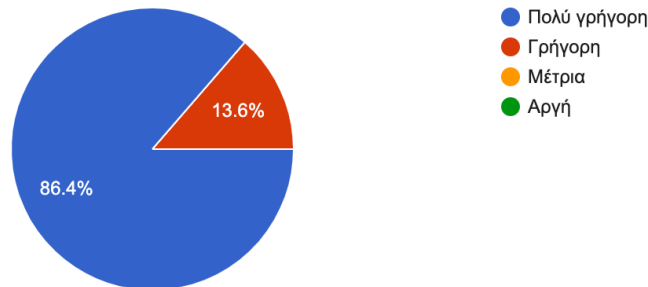
22 responses



Στις επόμενες δύο ερωτήσεις ζήτησα να σαρώσουν δυο Qr Code από δύο διαφορετικές σελίδες της εφαρμογής και να κρίνουν την ταχύτητα φόρτωσης της σελίδας για κάθε ένα από αυτά. Η πρώτη σελίδα περιείχε φωτογραφίες οι οποίες ήταν βελτιστοποιημένες για το διαδίκτυο ενώ η δεύτερη σελίδα περιείχε φωτογραφίες οι οποίες είχαν μεγάλο μέγεθος (Πάνω από 1MB). Τα αποτελέσματα των απαντήσεων ήταν τα αναμενόμενα μιας και για την πρώτη σελίδα το 86.4% απάντησε πως η φόρτωση του περιεχομένου ήταν **Πολύ Γρήγορη** ενώ το 13.6% πως ήταν **Γρήγορη**.

Σκανάρετε το παρακάτω QrCode με το κινητό σας (Dodoni Theatre). Η φόρτωση του περιεχομένου και των εικόνων ήταν:

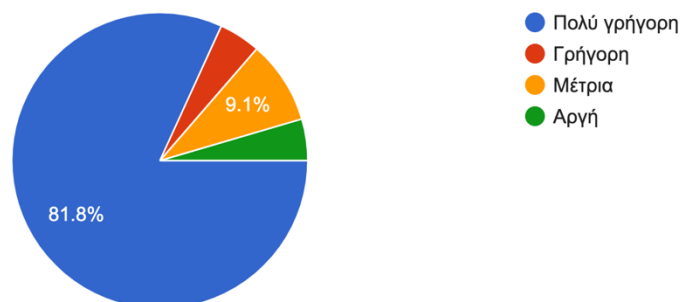
22 responses



Αντίθετα, για την δεύτερη σελίδα που περιείχε τις ογκώδεις φωτογραφίες το 81.8% απάντησε πως η φόρτωση της σελίδας ήταν **Πολύ Γρήγορη** ενώ το 18.2% απάντησε πως η φόρτωση του περιεχομένου της σελίδας ήταν **Μέτρια/Αργή**.

Σκανάρετε το παρακάτω QrCode με το κινητό σας (Aretaios). Η φόρτωση του περιεχομένου και των εικόνων ήταν:

22 responses

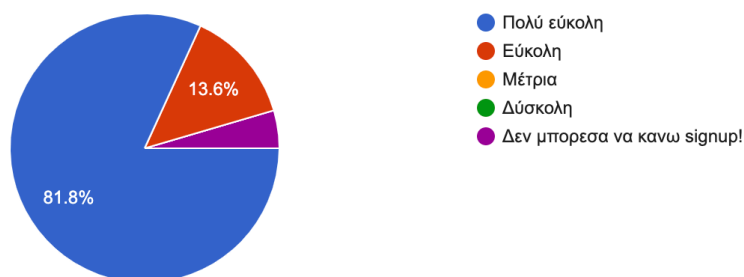


Το παραπάνω μικρό πείραμα με τις φωτογραφίες δείχνει πως ένα απλό λάθος στη δημιουργία μιας εφαρμογής μπορεί να αλλοιώσει την κρίση ενός χρήστη σχετικά με αυτή. Η γρήγορη φόρτωση ενός περιεχομένου είναι εξαιρετικά σημαντική για την φήμη μιας εφαρμογής. Έρευνες έδειξαν πως το μέσο εύρος προσοχής ενός ανθρώπου είναι πάρα πολύ μικρό και ολοένα και ελαττώνεται. Το 2000 το μέσο εύρος προσοχής ήταν 12 δευτερόλεπτα ενώ σήμερα έχει πέσει στα 8. Αυτό σημαίνει ότι όταν ένας άνθρωπος επισκέπτεται μια ιστοσελίδα μπορεί να χάσει το ενδιαφέρον του και να φύγει από αυτή μέσα σε μόλις 8 δευτερόλεπτα αν αυτή δεν φορτώσει έγκαιρα ή δεν βρει γρήγορα την πληροφορία που θέλει. Αυτό καθιστά την γρήγορη φόρτωση περιεχομένου και τη σωστή σχεδίαση της εφαρμογής (ειδικά για κινητά) πολύ σημαντική (6).

Οι επόμενες ερωτήσεις δεν ήταν υποχρεωτικές. Ζήτησα από τους χρήστες να δημιουργήσουν έναν λογαριασμό την εφαρμογή και να αξιολογήσουν την διαδικασία, και έπειτα να προσπαθήσουν να δημιουργήσουν ένα δικό τους κτίριο στην εφαρμογή. Δείχνοντας με αυτόν τον τρόπο πόσο εύκολο είναι από αναγνώστης κάποιος να γίνει δημιουργός μέσω της εφαρμογής QBI. Τα αποτελέσματα ήταν ικανοποιητικά μιας και οι περισσότεροι βρήκαν τη δημιουργία λογαριασμού **Εύκολη**. Ένας από αυτούς τους χρήστες επέλεξε την κατηγορία **Άλλο** γράφοντας πως δεν κατάφερε να δημιουργήσει λογαριασμό. Υπήρχε ένα πρόβλημα στο διακομιστή καθιστώντας αδύνατο για κάποιον χρήστη να δημιουργήσει κάποιο λογαριασμό εκείνη τη στιγμή. Το πρόβλημα λύθηκε γρήγορα, δείχνοντας όμως πως μια εφαρμογή είναι ένας ζωντανός οργανισμός που η συνεχή επιτήρηση/υποστήριξη και η πρόληψη των προβλημάτων είναι πολύ σημαντική για να αποφεύγονται τέτοια προβλήματα να καταλήγουν στον τελικό χρήστη.

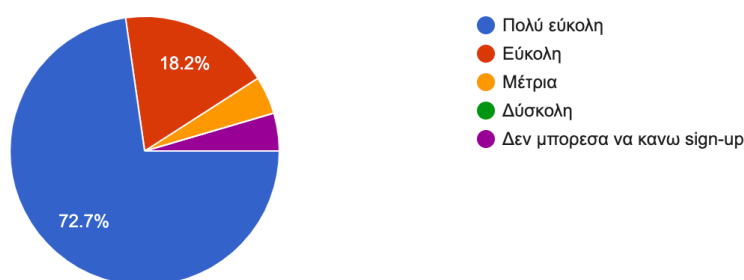
Δημιουργήστε έναν λογαριασμό στο QBI. Η διαδικασία δημιουργίας λογαριασμού ήταν:

22 responses

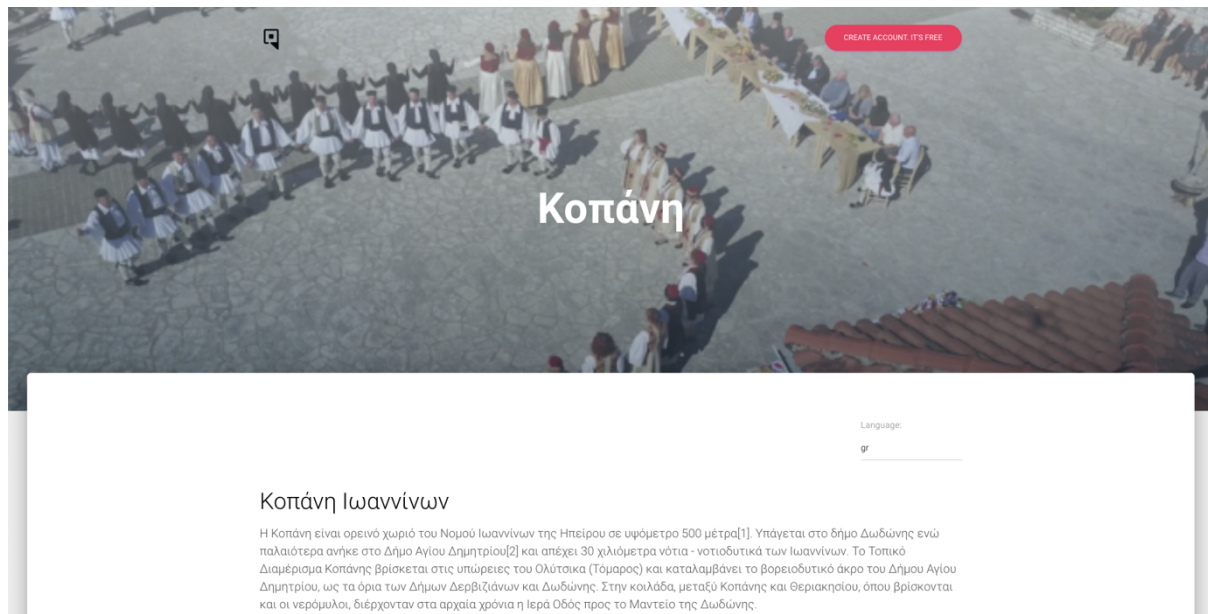


Με τον λογαριασμό που δημιουργήσατε, δοκιμάστε να προσθέσετε κάποια δική σας σελίδα. Η δημιουργία της δικής σας σελίδας ήταν:

22 responses

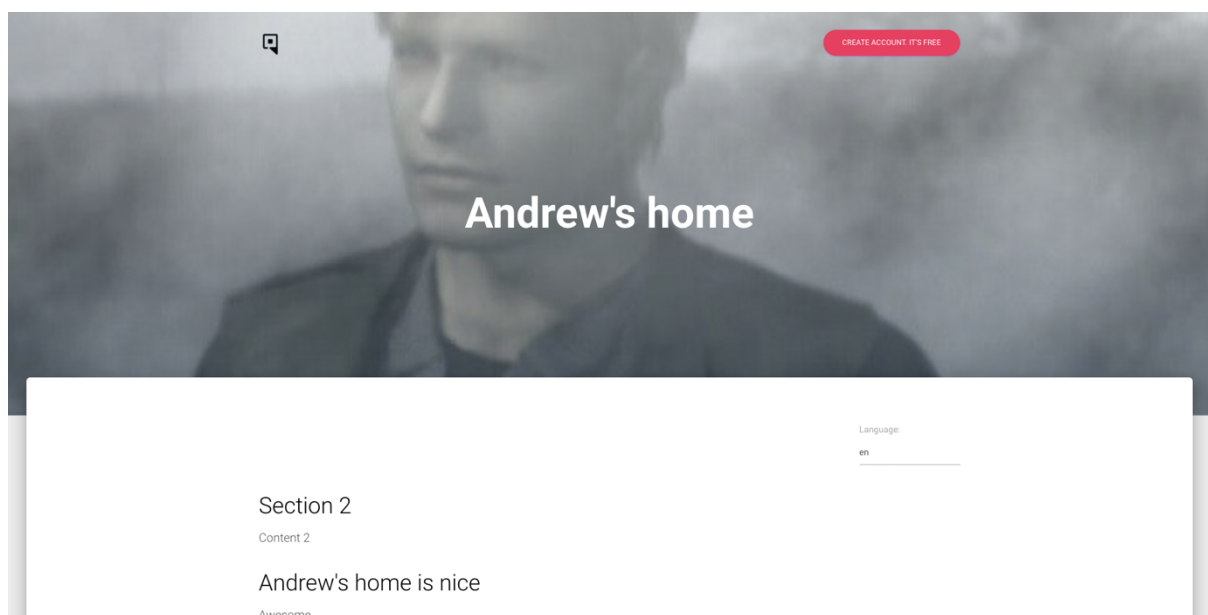


Από αυτές τις ερωτήσεις ένας από τους χρήστες επέλεξε να δημιουργήσει μια σελίδα σχετικά με ένα χωριό των Ιωαννίνων (Κοπάνη), δείχνοντας έτσι πώς μπορεί η πολιτισμική κληρονομιά και η Ιστορία να διανεμηθεί σε άλλους χρήστες μέσω της εφαρμογής.



Οι δυνατότητες που παρέχει αυτή η εφαρμογή στον καθένα να μπορέσει να προσθέσει πληροφορίες για κάποιο τόπο αναδεικνύουν την σημαντικότητά της στη διάχυση και τη διατήρηση της πολιτισμικής κληρονομιάς.

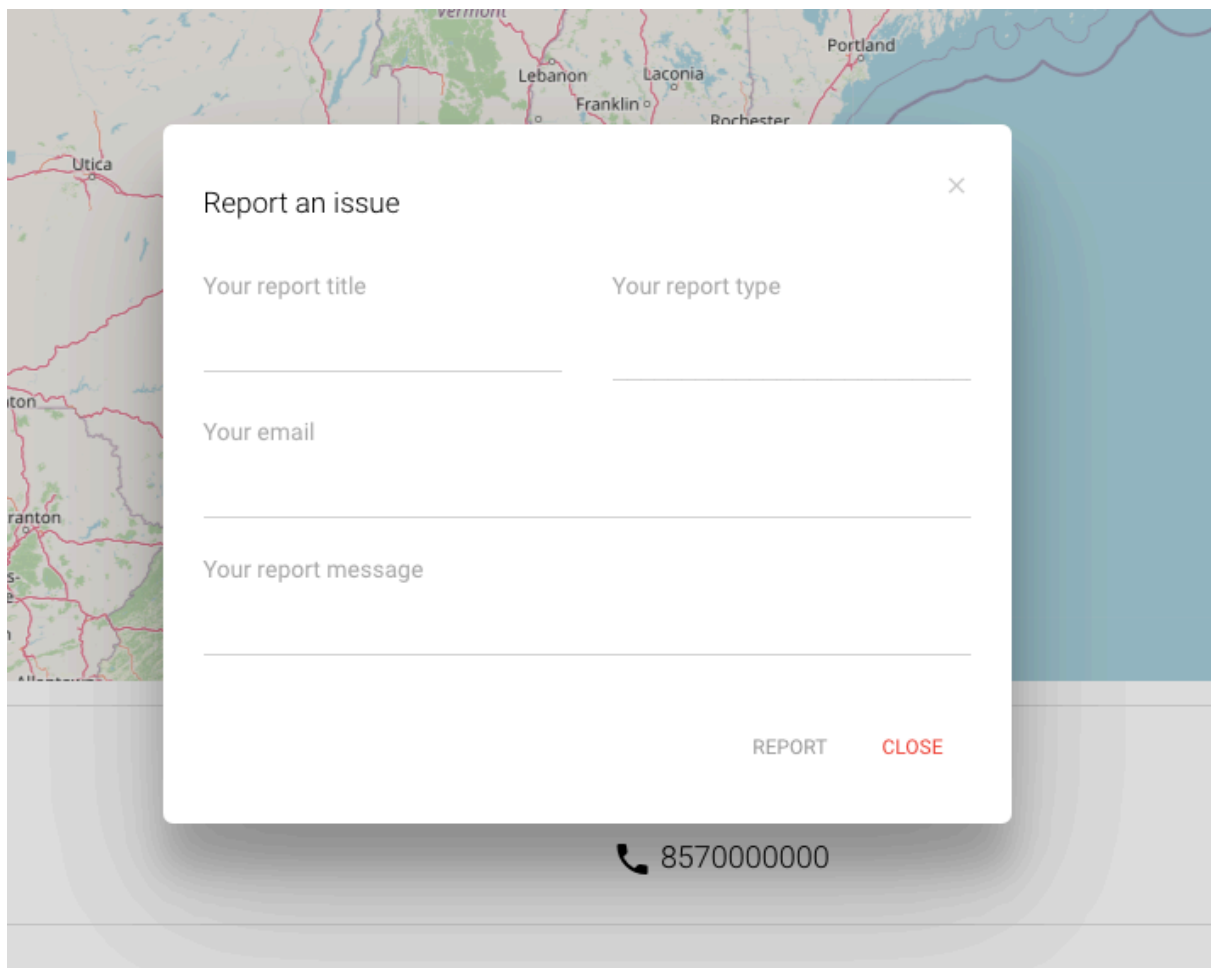
Τις δυνατότητες αυτές όμως ένας χρήστης μπορεί να τις εκμεταλλευτεί με κακόβουλο τρόπο. Όπως στο παρακάτω παράδειγμα όπου ένας άλλος χρήστης επέλεξε να δημιουργήσει μια σελίδα η οποία δεν έχει καμία απολύτως σχέση με Ιστορία ή Πολιτισμική κληρονομιά. Αυτό δείχνει πως ο καθένας μπορεί να δημιουργήσει περιεχόμενο ακόμη και αν αυτό είναι ανακριβές ή προσβλητικό.



Για αυτόν τον λόγο υλοποίησα την δημιουργία αναφοράς μιας σελίδας. Έτσι ένας χρήστης μπορεί να αναφέρει μια σελίδα η οποία περιέχει κάτι ανακριβές ή κακόβουλο προκειμένου οι συντηρητές της εφαρμογής να επιληφθούν του θέματος. Η αναφορά μιας σελίδας βρίσκεται στο κάτω μέρος της εκάστοτε προβολής όπως φαίνεται στις παρακάτω φωτογραφίες.

☎ 8570000000

⚠ REPORT AN ISSUE



The screenshot shows a mobile application interface. In the background, there is a map of Vermont with labels for cities like Utica, Lebanon, Franklin, Laconia, and Portland. Overlaid on the map is a white dialog box titled 'Report an issue' with a close button (X) in the top right corner. The dialog box contains four input fields: 'Your report title', 'Your report type', 'Your email', and 'Your report message'. At the bottom right of the dialog box, there are two buttons: 'REPORT' and 'CLOSE'.

Report an issue

Your report title

Your report type

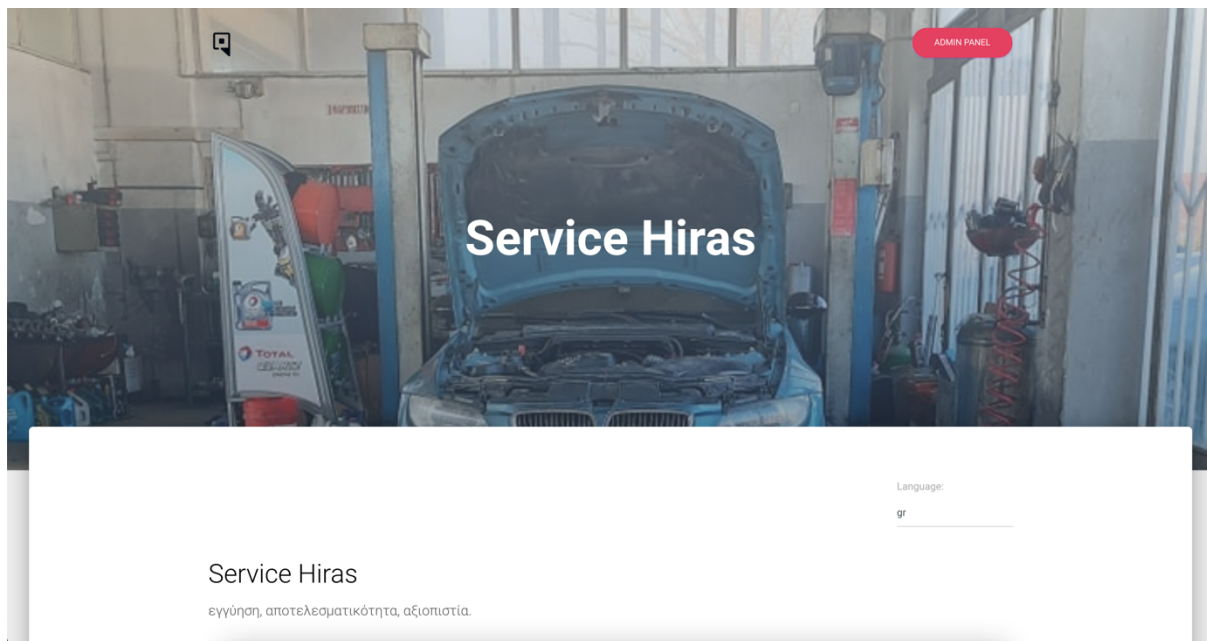
Your email

Your report message

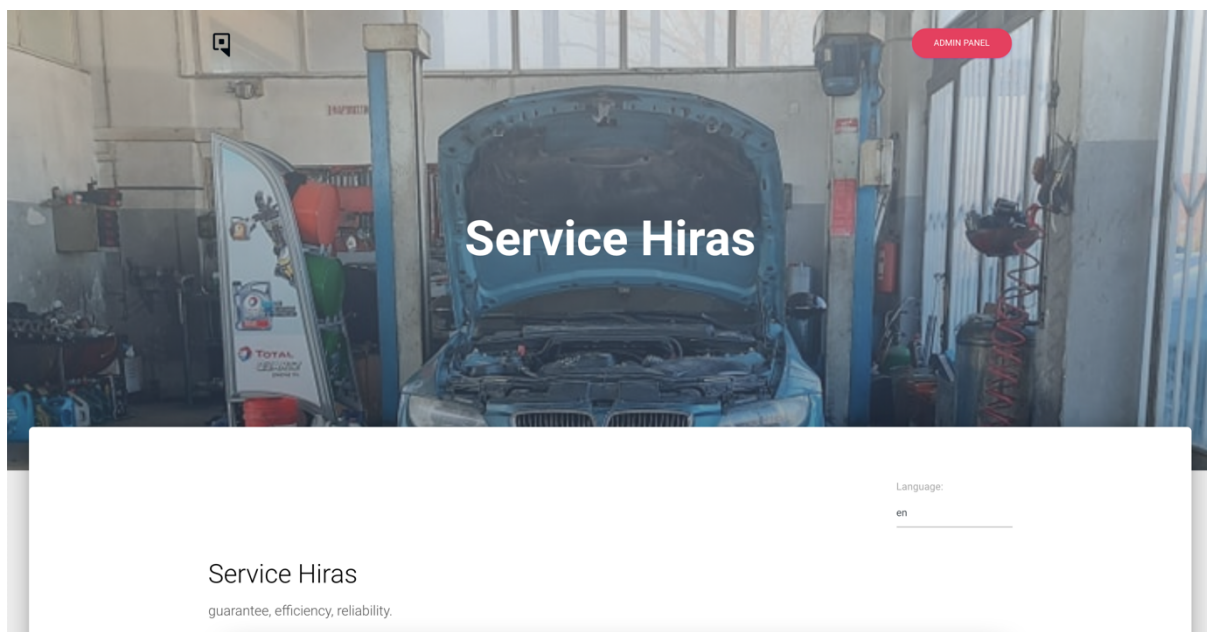
REPORT CLOSE

☎ 8570000000

Τέλος αξίζει να σημειωθεί πως ένας από τους χρήστες που πήραν μέρος στο ερωτηματολόγιο έκανε χρήση της υπηρεσίας όχι για τη δημιουργία κάποιας εγκατάστασης πολιτισμικής κληρονομιάς αλλά για ένα συνεργείο αυτοκινήτων. Δείχνοντας με αυτόν τον τρόπο ότι η πλατφόρμα μπορεί να χρησιμοποιηθεί όχι μόνο για πολιτισμικές εγκαταστάσεις αλλά και για εμπορικές. Έτσι κάποιος ο οποίος δεν θέλει να ασχοληθεί για την εύρεση και ενοικίαση διακομιστή ή την εύρεση συνεργάτη για την υλοποίηση της ιστοσελίδας, μπορεί πολύ απλά να χρησιμοποιήσει το QBI και να διανείμει με τέτοιο τρόπο τις πληροφορίες που θέλει σε πελάτες. Ο συγκεκριμένος χρήστης έκανε χρήση και τις λειτουργίες διαφορετικών γλωσσών που έχει το QBI, οπότε κάποιος χρήστης μπορεί να διαβάσει τις πληροφορίες σε διαφορετικές γλώσσες.



Η ελληνική έκδοχή της ιστοσελίδας του χρήστη



Η Αγγλική έκδοχή της ιστοσελίδας του χρήστη

Ιδέες επέκτασης

CI/CD (Συνεχής Ενσωμάτωση/Συνεχής ανάπτυξη)

Οποιαδήποτε εφαρμογή προκειμένου να αντέξει στο χρόνο πρέπει να αναπτύσσεται και να συντηρείται διαρκώς. Κάθε δημιουργία μιας νέας λειτουργίας θα πρέπει να είναι διαθέσιμη στον τελικό χρήστη με όσο πιο αυτοματοποιημένο τρόπο γίνεται και να γίνεται βέβαιο πώς η ένταξη μιας νέας λειτουργίας δεν θα χαλάσει κάποια άλλη. Για αυτό προτείνω σαν ιδέα επέκτασης την υλοποίηση του CI/CD μέσω του εργαλείου Jenkins.

Στη μηχανική λογισμικού, το CI/CD είναι συνδυασμένες πρακτικές συνεχούς παράδοσης και συνεχούς ανάπτυξης λογισμικού. Γεφυρώνει τα κενά μεταξύ των δραστηριοτήτων της ανάπτυξης και της λειτουργίας των ομάδων υλοποίησης λογισμικού επιβάλλοντας συγκεκριμένους αυτοματισμούς κατά την κατασκευή, τον έλεγχο και την ανάπτυξη εφαρμογών. Για την υλοποίηση των τεχνικών του CI/CD προτείνω την χρήση του εργαλείου Jenkins, που είναι ένα ανοιχτού κώδικα λογισμικό. Πρόκειται για έναν διακομιστή ο οποίος διαρκώς ελέγχει την εφαρμογή και κάθε νέο κομμάτι κώδικα που περνάει στο πυρήνα της εφαρμογής κάνοντας εύκολο για κάποιον ή για κάποια ομάδα να δίνει στον τελικό χρήστη πάντα την πιο σταθερή εκδοχή της εφαρμογής αποφεύγοντας λάθη κατά την υλοποίηση (7).

IoT (Beacon, Proximity)

Μια πολύ ενδιαφέρουσα προσθήκη στην εφαρμογή θα ήταν και το IoT (Internet of Things). Πρόκειται για την έννοια της διασύνδεσης φυσικών συσκευών – έξυπνων συσκευών με το διαδίκτυο. Αυτό σημαίνει πως ένα αντικείμενο μπορεί να ανιχνευθεί ή να εκλεχθεί εξ αποστάσεως μέσω της υπάρχουσας υποδομής δικτύου, επιτρέποντας έτσι την άμεση ενσωμάτωση του φυσικού κόσμου με το διαδίκτυο. Ένας τύπος μιας τέτοιας συσκευής είναι και το Beacon. Μια μικρή συσκευή (συνήθως Bluetooth) η οποία εκπέμπει ανά ένα χρονικό περιθώριο με μικρή γεωγραφική ακτίνα κάνοντας την συσκευή ανιχνεύσιμη από ένα κινητό τηλέφωνο το οποίο βρίσκεται κοντά στην περιοχή ανίχνευσής της. Για παράδειγμα, μπορεί μια τέτοια συσκευή να περιέχει ένα μοναδικό αναγνωριστικό με άμεση σύνδεση με μια σελίδα του QBI και έτσι όταν ένας χρήστης περάσει από ένα κτίριο ή περιοχή πολιτισμικής σημασίας το κινητό του μπορεί αυτόματα να τον ειδοποιήσει και αυτός να επιλέξει αν θέλει να διαβάσει πληροφορίες σχετικά με το συγκεκριμένο σημείο ή κτίριο. Αυτή η λειτουργία μπορεί να είναι επιπρόσθετη του Qr Code (10).

Εμπειρία 3D

Πλέον η υπολογιστική ισχύ των υπολογιστών αλλά και των κινητών τηλεφώνων επιτρέπουν την δημιουργία και προβολή εντυπωσιακών σκηνών 3D. Θα μπορούσε να υπάρχει επιλογή στην εφαρμογή η μεταφόρτωση περιεχομένου 3D όπου κάποιος χρήστης θα έχει τη δυνατότητα να δει έναν χώρο πολιτισμικής σημασίας και κληρονομιάς μέσω του QBI. Με αυτόν τον τρόπο κάποιος χρήστης θα μπορέσει να εξερευνήσει και να παρατηρήσει προσθέτοντας έτσι μια διαφορετική εμπειρία στην χρήση της εφαρμογής.

Βιβλιογραφία

1. 8 Proven Reasons you need angular for your next development project (Angular)
<https://www.grazitti.com/blog/8-proven-reasons-you-need-angular-for-your-next-development-project/>
2. Model-View-ViewModel (MVVM) Explained (MVVM)
<https://www.wintellect.com/model-view-viewmodel-mvvm-explained/>
3. How Single Sign On Works (SSO)
<https://www.onelogin.com/learn/how-single-sign-on-works>
4. Auth2 <https://oauth.net/2/>
5. Authentication and Authorization with OAuth (Pros and Cons)
<https://medium.com/@nimansaathukorala/authentication-and-authorization-with-oauth-1ff6776f2e6b>
6. The Human Attention Span
<https://www.digitalinformationworld.com/2018/09/the-human-attention-span-infographic.html#:~:text=A%20recent%20study%20found%20that,attention%20span%20of%20nine%20seconds.>
7. Continuous integration vs continuous delivery vs continuous deployment
<https://www.atlassian.com/continuous-delivery/principles/continuous-integration-vs-delivery-vs-deployment>
8. Information Dissemination through Official Websites: A Critique (Bhattacharya, Kaushik)
9. Information Dissemination through Web Portals (based on SOA)
10. Publishing and Using Cultural Heritage Linked Data on the Semantic Web
11. QR Codes in Libraries: New Delivery Tool for Library Marketing and Information Dissemination
12. Η πολιτιστική κληρονομιά και η ερμηνεία της στην ψηφιακή εποχή
13. The Little-Known Story of the Birth of the QR Code
<https://www.nippon.com/en/news/fnn20191214001/the-little-known-story-of-the-birth-of-the-qr-code.html>
14. Managing the Data-base Environment (James Martin, James Martin, S.J, James Thomas Martin)
15. How To Install the Apache Web Server on Ubuntu 18.04
<https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-18-04>
16. How To Install MySQL on Ubuntu 18.04
<https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-18-04>
17. How To Set Up a Node.js Application for Production on Ubuntu 16.04
<https://www.digitalocean.com/community/tutorials/how-to-set-up-a-node-js-application-for-production-on-ubuntu-16-04>
18. How To Set Up Apache Virtual Hosts on Ubuntu 18.04
<https://www.digitalocean.com/community/tutorials/how-to-set-up-apache-virtual-hosts-on-ubuntu-18-04>
19. Τι είναι ο πάπυρος <http://www.calligraphy-illumination.com/el/materials-techniques/papyrus>
20. Ο πάπυρος και η περγαμινή ως υλικά γραφής στον αρχαίο κόσμο
<https://geonews.gr/o-papyros-kai-h-pergamini-ws-ylika-grafis-ston-archaio-kosmo/>

21. A short history of the web <https://home.cern/science/computing/birth-web/short-history-web>

Γλωσσάρι Ξενικών Όρων

Qr Building Information

Qr Πληροφορίες εγκαταστάσεων/κτιρίων

Qr Code (Quick Response code)

Κωδικός γρήγορης απόκρισης

Software as a Service

Λογισμικό ως υπηρεσία

Application Programming Interface

Διεπαφή προγραμματισμού εφαρμογών

Web Browser

Πρόγραμμα περιήγησης ιστού

World Wide Web

Παγκόσμιος Ιστός

Hypertext Markup Language

Γλώσσα σήμανσης υπερκειμένου

Blog

Ιστολόγιο

Service Oriented Architecture

Αρχιτεκτονική Προσανατολισμού Υπηρεσιών

Web Application

Διαδικτυακή εφαρμογή

Client-Server

Χρήστης-Διακομιστής

File Transfer Protocol

Πρωτόκολλο μεταφοράς αρχείων

CRUD (Create, Read, Update, Delete)

Δημιουργία, Ανάκτηση, Ενημέρωση, Διαγραφή

REST API

Είναι ένα αρχιτεκτονικό στυλ λογισμικού το οποίο καθορίζει ένα σύνολο περιορισμών που πρέπει να ακολουθούνται για την δημιουργία υπηρεσιών διαδικτύου.

Design Patterns

Σχεδιαστικά πρότυπα λογισμικού

Repository Pattern, Single Responsibility Pattern, Open-Closed pattern , Dependency Inversion, Model-Route -Service, Observer Pattern

Τα παραπάνω πρόκειται για τύπους σχεδιαστικών προτύπων λογισμικού τα οποία αναλύονται εκτενώς στο κείμενο της πτυχιακής.

Private API

Πρόκειται για κρυφή διεπαφή λογισμικού, με την οποία γίνεται εφικτή η ασφαλής ανταλλαγή δεδομένων από την βάση δεδομένων στην εφαρμογή.

Public API

Πρόκειται για δημόσια διεπαφή λογισμικού, με την οποία ο οποιοσδήποτε μπορεί να αποκτήσει πρόσβαση σε συγκεκριμένες πτυχές της βάσης δεδομένων.

Frontend

Πρόκειται για την πρακτική μετατροπής δεδομένων σε γραφική διεπαφή, κάνοντας χρήση γλωσσών προγραμματισμού για το διαδίκτυο όπως html, css, JavaScript.

Framework

Στη μηχανική λογισμικού, ένα framework (πλαίσιο λογισμικού) είναι κομμάτια κώδικα χωρισμένα σε μια βιβλιοθήκη την οποία μπορεί κάποιος προγραμματιστής να χρησιμοποιήσει για να λύσει κάποιο πρόβλημα ή να υλοποιήσει μια εφαρμογή.

SSO (Single Sign On)

Η μοναδική σύνδεση είναι ένα σχήμα ελέγχου ταυτότητας που επιτρέπει σε έναν χρήστη να συνδεθεί με ένα μόνο αναγνωριστικό σε οποιοδήποτε από τα πολλά σχετικά, αλλά ανεξάρτητα, συστήματα λογισμικού.

Token

Στη μηχανική λογισμικού, το Token περιέχει διαπιστευτήρια ασφαλείας για μια περίοδο σύνδεσης και προσδιορίζει τα δικαιώματα κάποιο χρήστη για κάποια συγκεκριμένη εφαρμογή.

Edit

Επεξεργασία

View

Προβολή

Virtual Host

Συχνά συντομευμένο και ως VHost, είναι ένας εικονικός πάροχος υπηρεσιών διαδικτύου που περιλαμβάνει λειτουργίες διακομιστή και υπηρεσίες σύνδεσης στο διαδίκτυο.

Domain Name System

Σύστημα ονομάτων τομέα, είναι το αναγνωριστικό μιας εφαρμογής στο διαδίκτυο.

FTP (File Transfer Protocol)

Το πρωτόκολλο μεταφοράς αρχείων είναι ένα τυπικό πρωτόκολλο δικτύου που χρησιμοποιείται για τη μεταφορά αρχείων υπολογιστή μεταξύ πελάτη και διακομιστή σε ένα δίκτυο υπολογιστών.

Quality Assurance Engineers

Μηχανικοί Διασφάλισης Ποιότητας