

ΑΣΚΗΣΗ ΑΞΙΟΛΟΓΗΣΗΣ ΕΠΑΡΚΕΙΑΣ ΜΗΧΑΝΙΚΩΝ ΛΟΓΙΣΜΙΚΟΥ

WEBAPP v3.2

Διαδικασία

Περιγραφή

Σκοπός της εργασίας είναι να φτιαχτεί ένα πλήρες web application που να προσφέρει τις παρακάτω λειτουργικές δυνατότητες:

- Ορισμό κατηγορίας χαρακτηριστικών (attributes).
- Ορισμό Εργαζομένων (employees).
- Ανάθεση Χαρακτηριστικών σε Εργαζομένους.
- Παρουσίαση θέσης εργαζομένων.
- Δρομολόγηση μεταξύ εργαζομένων.

Για την λειτουργία του Web Application θα απαιτηθεί να κατασκευαστούν:

- Η βάση δεδομένων σε σύστημα που υποδεικνύεται.
- Το υπόστρωμα υπηρεσιών (backend business logic / data access etc).
- Η προγραμματιστική διεπαφή (REST API)
- Η γραφική διεπαφή χρήστη.

Τεχνολογικές Επιλογές

Σε κάθε σχεδόν σημείο υλοποίησης, προτείνονται διάφορες εναλλακτικές τεχνολογικές πλατφόρμες, οι οποίες επιτρέπουν την επίτευξη του επιθυμητού αποτελέσματος. Σε κάποιες περιπτώσεις υποδεικνύεται η σειρά προτεραιότητας την οποία θέτει η Εταιρεία ως προς την εξοικείωση, όμως όλες οι άλλες προσεγγίσεις είναι αποδεκτές. Περαιτέρω όμως απόκλιση (πχ μη επιλογή μίας από τις τεχνολογίες) θα πρέπει να τεκμηριώνεται επαρκώς ώστε να γίνεται αποδεκτή προς αξιολόγηση.

Διαδικασία εκτέλεσης / Αξιολόγησης

Κατά τη φάση της αξιολόγησης εξετάζονται μόνο πλήρεις υλοποιήσεις. Βαρύτητα έχει ο χρόνος υλοποίησης, αλλά μεγαλύτερη ακόμα σημασία έχει η ποιότητά της, οπότε συνιστάται να μην προσπαθήσετε να την στείλετε όσο πιο γρήγορα μπορείτε έχοντας ως μόνο κριτήριο το χρόνο.

- Όταν παραλάβετε την άσκηση θα πρέπει να προσδιορίσετε τον χρόνο που θα χρειαστείτε ώστε να την εκτελέσετε. Η ομάδα αξιολόγησης λαμβάνει υπόψη σε κατά το δυνατόν τις ιδιαίτερες συνθήκες υπό τις οποίες κάποιος υποψήφιος καλείται να εξεταστεί (πχ εργασία, φοίτηση κ.α.), όμως σε κάθε περίπτωση μην παραλείψετε να επισημάνετε ειδικούς περιορισμούς που μπορεί να έχετε κατά την υποβολή της εκτίμησής σας. Ταυτόχρονα, αν δεν σας έχουν δοθεί σαφείς τεχνολογικές κατευθύνσεις για την υλοποίηση, θα πρέπει να περιλάβετε στην απάντησή σας και τις τεχνολογίες που θα χρησιμοποιήσετε.
- Αφού αναλυθεί η διάρκεια της αρχικής εκτίμησής σας και των τεχνολογιών που επιλέξατε, θα λάβετε το σχετικό μήνυμα ενημέρωσης, το οποίο, είτε θα επιβεβαιώνει το ενδιαφέρον της ομάδας αξιολόγησης, είτε θα ζητά περισσότερες λεπτομέρειες ή τροποποιήσεις επιλεγμένων τεχνολογιών, είτε θα αποσύρει το

ενδιαφέρον της. Εφόσον επιβεβαιώνεται το ενδιαφέρον, τότε θα μπορείτε να προχωρήσετε στην εκτέλεση της άσκησης.

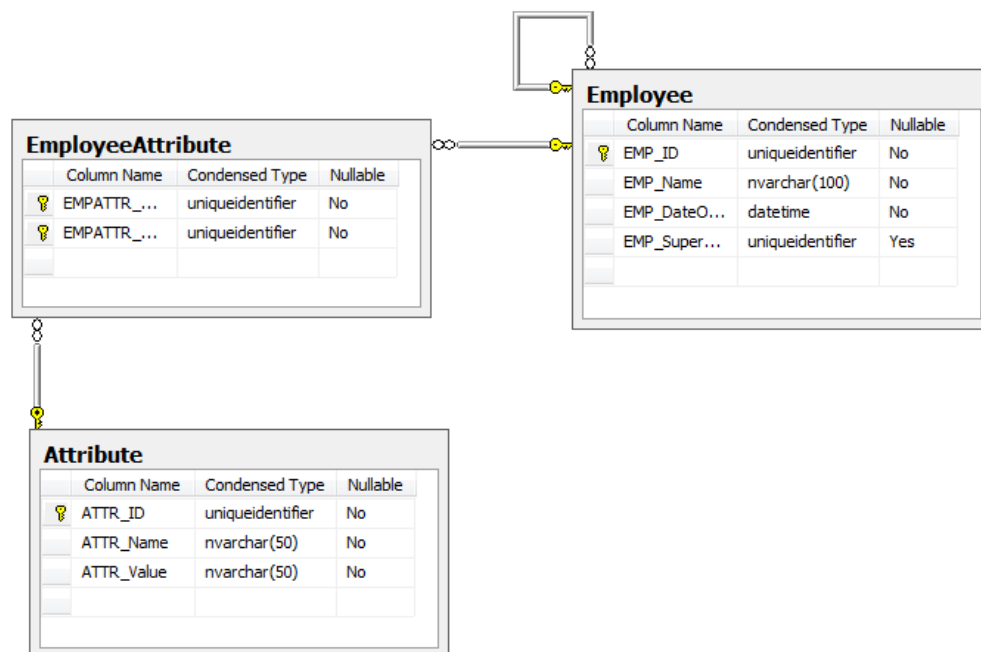
- Όταν ολοκληρώσετε την άσκηση, μέσα στο χρόνο που αρχικά εκτιμήσατε ότι απαιτείται, στέλνετε τα αποτελέσματα μέσω email και αναμένετε επιβεβαίωση λήψης. Αν ολοκληρώσετε την άσκηση με μικρή απόκλιση, θα πρέπει να τεκμηριώσετε τα προβλήματα που σας ανάγκασαν να αποκλίνετε από την αρχική σας εκτίμηση.
- Εφόσον ο χρόνος και η ποιότητα εκτέλεσης καλύπτουν τις ελάχιστες απαιτήσεις της CITE, θα κληθείτε σε τελική συνέντευξη επιβεβαίωσης του αποτελέσματος και της κατανόησης.

Σε κάθε σημείο (από την παραλαβή της άσκησης μέχρι την ολοκλήρωσή της) μην διστάσετε να επικοινωνήσετε με την Εταιρεία μας για οποιαδήποτε διευκρίνιση ή αλλαγή του προγραμματισμού σας.

Η Άσκηση

Μοντέλο Δεδομένων Εφαρμογής

Το παρακάτω σχήμα δείχνει τις οντότητες και τις αντίστοιχες συσχετίσεις τους:



Η Βάση δεδομένων θα πρέπει να υλοποιηθεί σε κατά προτίμηση ένα από τα παρακάτω ευρέως διαδεδομένα RDBMS (χωρίς σειρά προτίμησης):

- Microsoft SQL Server
- MariaDB / MySQL
- PostgreSQL

Θα πρέπει να παραδώσετε και να περιγράψετε τον σεναριοκώδικα δημιουργίας της ΒΔ ο οποίος επίσης θα εξεταστεί. Δεν είναι όμως απαραίτητο να δημιουργήσετε τις οντότητες με σεναριοκώδικα, αλλά μπορείτε να χρησιμοποιήσετε τα σχετικά γραφικά περιβάλλοντα των RDBMSes που αναφέρονται παραπάνω (πχ SQL Server Manager Studio, PG Admin κοκ). Σκοπός του μέρους αυτού της άσκησης είναι να αναγνωρίσετε τις όποιες ελλείψεις και να υλοποιήσετε ένα ικανό μοντέλο ώστε να αποδοθεί η ζητούμενη παρακάτω λειτουργικότητα της εφαρμογής.

Λειτουργικότητα Εφαρμογής

Από πλευράς οθονών και διαδικασίας χρήσης της εφαρμογής, θα πρέπει να παρέχονται τα παρακάτω:

- Ένα κεντρικό view από όπου θα παρέχεται πρόσβαση σε υπο-οθόνες. Οι υπο-οθόνες θα είναι οι παρακάτω:
 - Attributes
 - Employees
 - Map
- Η οθόνη Attributes θα παρέχει
 - Λίστα των υπαρχόντων attributes
 - Υπάρχει κουμπί «add» που επιτρέπει τη δημιουργία νέου attribute
 - Πατώντας το κουμπί add, ο χρήστης οδηγείται σε νέα οθόνη όπου μπορεί να εισάγει το όνομα του attribute
 - Πατώντας σε υπάρχον attribute ο χρήστης οδηγείται σε view επεξεργασίας όπου επιτρέπεται η μετονομασία και η διαγραφή του χαρακτηριστικού
 - Σε περίπτωση διαγραφής, αφαιρείται η σύνδεση του χαρακτηριστικού από όλους τους συνδεδεμένους εργαζομένους
- Η οθόνη Employees θα παρέχει
 - Λίστα των υπαρχόντων employees
 - Υπάρχει κουμπί «add» που επιτρέπει τη δημιουργία νέου employee.
 - Πατώντας το κουμπί add, ο χρήστης οδηγείται σε νέα οθόνη όπου μπορεί να εισάγει τα στοιχεία του εργαζομένου
 - Πατώντας σε υπάρχον employee ο χρήστης οδηγείται σε view επεξεργασίας όπου επιτρέπεται η επεξεργασία των στοιχείων του εργαζομένου και η διαγραφή του
 - Μέσω της οθόνης δημιουργίας / επεξεργασίας του εργαζομένου τα εξής στοιχεία μπορούν να εισαχθούν / επεξεργαστούν
 - Όνομα εργαζομένου
 - Ημερομηνία γέννησης εργαζομένου
 - Αν διαθέτει αυτοκίνητο
 - Η διεύθυνση κατοικίας του
 - Η λίστα των attributes που τον χαρακτηρίζει
 - Τα διαθέσιμα attributes είναι αυτά που έχουν οριστεί μέσω της αντίστοιχης οθόνης των attributes
 - Σε αυτό το component ο χρήστης μπορεί να αφαιρέσει υπάρχοντα attributes ή να προσθέσει νέα.
 - Κάθε εργαζόμενος μπορεί να έχει μόνο μία σύνδεση με κάθε attribute
- Η οθόνη του χάρτη έχει την εξής λειτουργικότητα
 - Αναζήτηση και επιλογή εργαζομένου
 - Σαν κριτήριο αναζήτησης, ο χρήστης μπορεί να επιλέξει από μία λίστα ένα χαρακτηριστικό που θέλει να έχουν οι εργαζόμενοι από τη λίστα των attributes όπως αυτά δημιουργήθηκαν από την αντίστοιχη οθόνη των attributes
 - Εμφανίζεται η λίστα των εργαζομένων που ικανοποιούν τα κριτήρια
 - Ο χρήστης επιλέγει έναν από τους εργαζόμενους που εμφανίζονται
 - Πατάει το κουμπί «Continue»
 - Πηγαίνει σε επόμενη οθόνη δρομολόγησης
 - Δρομολόγηση εργαζομένου
 - Στην οθόνη αυτή εμφανίζεται ένα χάρτη της google

- Πάνω στον χάρτη εμφανίζονται με pins όλοι οι employees που έχουν οριστεί από την αντίστοιχη οθόνη εργαζομένων βάσει διεύθυνσης κατοικίας
- Ο εργαζόμενος που επιλέχθηκε από το προηγούμενο βήμα αναζήτησης και επιλογής εμφανίζεται με διαφορετικό pin
- Εμφανίζεται στο χάρτη μια διαδρομή που συνδέει τον επιλεγμένο εργαζόμενο με όλους τους υπόλοιπους εργαζομένους
- Ανάλογα με το αν ο επιλεγμένος εργαζόμενος έχει στα στοιχεία του ορισμένο ότι διαθέτει αυτοκίνητο ή όχι, η διαδρομή που εμφανίζεται θα πρέπει να είναι για μετακίνηση μέσω αυτοκινήτου ή οδοιπορικώς
- Σημείωση για το routing και την τοποθέτηση των εργαζομένων στον χάρτη
 - Ανάλογα με το διαθέσιμο χρόνο ολοκλήρωσης και τη διαθεσιμότητα των αντίστοιχων Google API δίνονται οι παρακάτω επιλογές:
 - Στα στοιχεία του εργαζομένου, αντί για διεύθυνση, μπορεί να γίνει εισαγωγή συντεταγμένων της κατοικίας του εργαζομένου
 - Στη δρομολόγηση, αντί για routing βάσει διαδρομής και είδους μετακίνησης, μπορεί να γίνει σχεδίαση διαδοχικών γραμμών σύνδεσης ασχέτως διαδρομής / δρόμων / είδους μετακίνησης. Τα edges θα συνδέουν απλά τις θέσεις των εργαζομένων

Κατευθύνσεις υλοποίησης

- Η εφαρμογή θα πρέπει να δομηθεί σε 4 διακριτά layers, εκτός αν κάποιο από αυτά ρητά εξαιρείται μετά από οδηγία που σας έχει δοθεί:
 - Βάση Δεδομένων
 - Data Access / Business Layer
 - Web API – http endpoints / RESTful
 - Web App – Client-side rendering
- Για το Web App δεν επιτρέπεται η χρήση server side rendering τεχνολογιών.
- Προτείνονται σαν βιβλιοθήκες / framework προς χρήση κατά σειρά προτεραιότητας οι εξής, εφόσον δεν σας έχει υποδειχθεί κάποια συγκεκριμένη:
 1. Angular
 2. React
 3. JQuery
 4. Vanilla JavaScript
 5. Άλλες τεχνολογίες μπορούν να χρησιμοποιηθούν αλλά πρέπει να αιτιολογείται η χρήση τους
- Η εφαρμογή Web App θα πρέπει να ακολουθεί λογική Single Page Application (SPA)
 - Εξαίρεση αποτελεί η περίπτωση που σας έχει υποδειχθεί ρητά η χρήση Plain JS/HTML/CSS και όχι δημιουργίας SPA.
- Θα πρέπει να γίνεται validation των δεδομένων εισόδου χρήστη τόσο στο frontend όσο και σε backend με αντίστοιχα μηνύματα λαθών.
 - Εξαίρεση αποτελεί η περίπτωση που σας έχει υποδειχθεί ρητά η χρήση Plain HTML/CSS/JS και όχι δημιουργίας SPA, οπότε το validation δεν θα γίνεται στο front-end αλλά θα μεταφέρονται σε αυτό τα σφάλματα.
- Το σύνολο του styling θα πρέπει να γίνεται μέσω κατάλληλων CSS files
 - UI libraries / guidelines όπως τα παρακάτω μπορούν να χρησιμοποιηθούν

- Bootstrap
 - Material for Angular
 - JQuery UI
- Άλλες τεχνολογίες μπορούν να χρησιμοποιηθούν αλλά πρέπει να αιτιολογείται η χρήση τους
- Το σύνολο της ανάκτησης δεδομένων και ενεργειών να γίνεται μέσω κατάλληλων web API calls. Η εφαρμογή θα πρέπει να επιδεικνύει την χρήση AJAX calls.
 - Εξαίρεση αποτελεί η περίπτωση που σας έχει υποδειχθεί ρητά η χρήση Plain HTML/CSS και όχι δημιουργίας SPA.
- Το Web API μπορεί να υλοποιηθεί, ανάλογα με την επιλεγόμενη τεχνολογία που σας έχει υποδειχθεί, σε:
 - ASP.NET Core
 - Spring Boot
- Για το Web API θα πρέπει να γίνει κατάλληλη χρήση των αντίστοιχων τεχνικών IoC και Dependency Injection.
- Για το Data Access, ανάλογα την επιλεγόμενη τεχνολογία, μπορεί να χρησιμοποιηθεί κατά σειρά προτεραιότητας
 - JPA ή Entity Framework Core
 - Raw SQL μόνο κατ' εξαίρεση και μετά από επικοινωνία με ή προτροπή από την ομάδα αξιολόγησης.
- Για την εφαρμογή του μοντέλου δεδομένων δεν επιτρέπεται η χρήση code first approach και οι οντότητες πρέπει να δημιουργηθούν χωρίς χρήση εργαλείων αυτοματοποίησης.
- Αν υλοποιείτε Data Access / Business Layer, και σας έχει ζητηθεί η χρήση Design Patterns, θα πρέπει να κάνετε ρητή επίδειξη κατανόησης και εφαρμογής ευρέως διαδεδομένων όπως Dependency Injection, Factory, Visitor, Observer, Singleton κ.α. Η αναφερόμενη εφαρμογή τους, δεν αναφέρεται στην χρήση design patterns την οποία κάνει το ίδιο το framework ανάπτυξης που χρησιμοποιείτε, αλλά σε δικό σας σχεδιασμό και εκμετάλλευσή τους για τις ανάγκες της επιχειρησιακής λογικής σας.

Καλή επιτυχία!
