

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΣΥΓΧΡΟΝΑ ΘΕΜΑΤΑ ΤΕΧΝΟΛΟΓΙΑΣ ΛΟΓΙΣΜΙΚΟΥ

ΤΕΧΝΙΚΟ ΕΓΧΕΙΡΙΔΙΟ ΕΡΓΑΣΙΑΣ ΜΥС

ΟΜΑΔΑ ΑΝΑΠΤΎΞΗΣ:

- ΝΙΚΟΛΑΣ ΠΑΤΕΡΑΣ Π17172
- ΑΝΔΡΕΑΣ ΘΕΟΔΩΡΙΔΗΣ **Π17164**
- ΒΑΣΙΛΕΙΟΣ ΖΑΡΤΗΛΑΣ ΠΑΠΑΧΑΡΑΛΑΜΠΟΥΣ Π17168

Πειραιάς, Ιανουάριος 2021

ПЕРІЕХОМЕНА

ΠΕΡΙΕΧΟΜΕΝΑ	2
ΕΙΣΑΓΩΓΗ	. 4
ΣΤΟΧΟΙ	. 4
ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ	. 5
MODELS	. 6
ModelPubs.edmx	. 6
PartialClasses.cs	. 6
\Metadata	. 6
CONTROLLERS	. 7
VIEWS	. 8
\References	. 8
Index.cshtml	. 8
ContactDetails.cshtml	. 8
OrderID.cshtml	. 8
\Jobs	. 9
Create.cshtml	9
Delete.cshtml	9
Edit.cshtml	. 9
Details.cshtml	. 9
Index.cshtml	10
\Shared	10
_Layout.cshtml	10
Global.asax.cs	10
\Error	10

NotFound.cshtml	10
ANAZHTHΣH ΣΕ ΠΙΝΑΚΑ ΚΑΙ ΧΡΗΣΗ PAGED LIST	
Αναζήτηση:	11
Paged List:	11
ΒΙΒΛΙΟΓΡΑΦΙΑ	12
ΡΔΟΚΔGES/ΕΡΓΔΛΕΙΔ	13

ΕΙΣΑΓΩΓΗ

Αρχικά, το **ASP.NET MVC** είναι ένα web application framework που δημιουργήθηκε από την Microsoft και υλοποιεί το design pattern model–view–controller (**MVC**) pattern. **Δεν συντηρείται πλέον** και είναι open-source λογισμικό. Από τότε το **ASP.NET Core** έχει κυκλοφορήσει, που έχει ενσωματώσει τα ASP.NET, ASP.NET MVC, ASP.NET Web API και ASP.NET Web Pages (μια πλατφόρμα που χρησιμοποιεί μόνο σελίδες «Razor»).

Το μοντέλο ΜVC καθορίζει τις εφαρμογές ιστού σε τρία λογικά επίπεδα:

- Model (επιχειρηματικό επίπεδο)
- View (επίπεδο προβολής)
- Controller (Έλεγχος Input)

ΣΤΟΧΟΙ

Υλοποιώντας μια εφαρμογές ιστού με ASP .NET MVC παρέχει πολλά οφέλη αφού μπορούμε να δημιουργήσουμε Cross platform εφαρμογές σε διάφορα λειτουργικά συστήματα. Επίσης, μπορούμε να ενσωματώνουμε JavaScript ευκολότερα.

Επίσης χρησιμοποιήσαμε κλάσεις bootstrap για την καλύτερη σχεδίαση της σελίδας μας!

To source code του project είναι διαθέσιμο στο GitHub αφού χρησιμοποιήσαμε version control.

https://github.com/IHateSyntaxErrors/MVCBookshelf

 $EI\Sigma A \Gamma \Omega \Gamma H$ 4

ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Συνδέσαμε τον server της βάσης με την public IP address ενός υπολογιστή και έτσι η υπόλοιπη ομάδα συνδεόταν μέσω της public IP στον server και μπορούσε να διαχειρίζεται όλα τα δεδομένα που υπήρχαν στην βάση pubs.

 $BA\Sigma H \Delta E\Delta OMEN\Omega N$ 5

MODELS



ModelPubs.edmx

Στο αρχείο αυτό εμφανίζεται το διάγραμμα της Βάσης Δεδομένων μας. Το διάγραμμα δημιουργήθηκε αυτόματα κατά την δημιουργία και την εισχώρηση της βάσης στο project μας.

PartialClasses.cs

Η κλάση αυτή μας βοηθά να αντιστοιχούμε τις metadata κλάσεις μας στις αντίστοιχες κλάσεις στις όποιες τις χρησιμοποιούμε. Για παράδειγμα μέσω της PartialClasses.cs αντιστοιχούμε την κλάση JobsMetadata.cs στην κλάση jobs.cs. Άρα εξασφαλίζουμε ότι η κλάση JobsMetadata αναφέρεται στην κλάση Jobs, κάνουμε δηλαδή extents τα attributes.

.

■ \Metadata

Μέσω του DisplayNameFor εμφανίζουμε τα αντίστοιχα ονόματα των attributes για τις στήλες στους πίνακες μας σε κάθε κλάση. Για παράδειγμα η γραμμή:

@Html.DisplayNameFor(model => model.First().job_desc)

θα εμφανίσει στον χρήστη το attribute job_desc της κλάσης Jobs ως το όνομα της αντίστοιχης στήλης του πίνακα.

Αυτό μπορούμε να το αλλάξουμε μέσω των Metadata.

Τα metadata είναι δεδομένα που περιγράφουν/αντικαθιστούν άλλα δεδομένα, ένα σύνολο metadata περιγράφει ένα σύνολο δεδομένων. Εμείς χρησιμοποιήσαμε metadata για να μπορέσουμε να παρέχουμε τα δεδομένα των πληροφοριών που βρίσκονται στη βάση σε φιλική μορφή προς τον χρήστη. Με άλλα λόγια μετονομάζουμε μεταβλητές μέσω των metadata. Παράδειγμα αν μία στήλη του πίνακα μας ονομάζεται stor_name στην βάση δεδομένων μας μέσω των metadata όπου εμφανίζεται η μεταβλητή stor_name στην ιστοσελίδα θα μετονομάζεται σε NAME STORE αυτόματα χωρίς να χρειαστεί να την αλλάξουμε εμείς hard-core.σσ

MODELS 6

ΕΡΓΑΣΙΑ ΜΥС

CONTROLLERS

Για να μπορέσουμε να εμφανίσουμε στις ανάλογες σελίδες τα στοιχεία από την βάση, χρησιμοποιήσαμε controllers για να τα εμφανίσουμε σε πίνακες για να μπορέσουμε να παρέχουμε στον χρήστη μια καλύτερη αλληλεπίδραση. Επίσης με τους controllers μπορούμε να καθορίσουμε την απάντηση που θα λάβει ο χρήστης για κάθε επιλογή που θα κάνει. Όπως για παράδειγμα έχουμε τον AuthorController που εμφανίζει σε ένα πίνακα τα στοιχεία του κάθε Author και επίσης μπορεί να προσθέσει ή να αφαιρέσει έναν Author.

CONTROLLERS 7

ΕΡΓΑΣΙΑ ΜΥС

VIEWS



\References



Index.cshtml

Στο πιο πάνω αρχείο βρίσκεται ο κώδικας html για την αρχική σελίδα. Επίσης, μετά από κάθε πάτημα ενός κουμπιού ο χρήστης μεταφέρεται σε μία νέα σελίδα ανάλογα με την ονομασία του κουμπιού όπως για παράδειγμα πατώντας το κουμπί "CONTACT DETAILS OF AUTHORS" εμφανίζει η φόρμα που μπορεί ο χρήστης να ψάξει ποιον Author θέλει με βάση τις περισσότερες πωλήσεις σε κάποιο χρονικό διάστημα. Στην αρχική σελίδα υπάρχουν δύο κουμπιά για να μεταφέρεται ο χρήστης στις φόρμες για την αναζήτηση.

ContactDetails.cshtml

Το πιο πάνω αρχείο είναι υπεύθυνο στο να εμφανίσει στον χρήστη το αποτέλεσμα της επιλογής που έχει κάνει στην φόρμα για να αναζητήσει για παράδειγμα τα στοιχεία επικοινωνίας ενός Author, «contact details of authors"», μπορεί να αφήσει κενά τα πεδία με τις ημερομηνίες αλλά όχι το πεδίο «Top sellers position».

Για παράδειγμα έχουμε δύο urls με τα στοιχεία που έχει δώσει ο χρήστης για να γίνει η αναζήτηση:

Στο πρώτο βλέπου ότι έχει συμπληρώσει όλα τα πεδία ενώ στο δεύτερο έχει αφήσει κενό το πεδίο «DateTo».

https://localhost:44356/References/ContactDetails?TopNum=5&DateFrom=1993-01-01&DateTo=1994-12-31

https://localhost:44356/References/ContactDetails?TopNum=5&DateFrom=1993-01-01&DateTo=



OrderID.cshtml

Το αρχείο αυτό λειτουργεί όπως το αρχείο ContactDetails.cshtml. Εδώ όμως αναζητά «Order ID Number». Πιο κάτω βλέπου την αντίστοιχη url για την φόρμα μας:

https://localhost:44356/References/OrderID?DateFrom=1993-01-01&DateTo=&StoreNameFrom=&StoreNameTo=

VIEWS 8



■ \Jobs

Στον φάκελο Jobs βρίσκονται τα παρακάτω αρχεία για τον πίνακα Jobs. Να συμπληρώσουμε ότι υπάρχει αντίστοιχος φάκελος για κάθε πίνακα της βάσης δεδομένων μας με τις ίδιες λειτουργείες. Να αναφέρουμε ότι με όμοιο τρόπο έγιναν όλα τα παρακάτω αρχεία για όλους τους πίνακες και σελίδες.



Create.cshtml

Στο αρχείο Create εμφανίζεται η σελίδα με τα αντίστοιχα πεδία κάθε πίνακα (labels και textboxes). Εδώ ο χρήστης μπορεί να προσθέσει μία πλειάδα στον πινακα εφόσον συμπληρώσει ορθά τα στοιχεία που του ζητούνται.



Delete.cshtml

Στο αρχείο Delete εμφανίζεται η σελίδα με τα στοιχεία τις κάθε πλειάδας που έχει επιλέξει ο χρήστης. Εδώ ο χρήστης μπορεί να διαγράψει μόνιμα την εκάστοτε πλειάδα.



Edit.cshtml

Στο αρχείο Edit εμφανίζεται η σελίδα με τα στοιχεία τις κάθε πλειάδας που έχει επιλέξει ο χρήστης. Εδώ ο χρήστης μπορεί να τα επεξεργαστεί και να αποθηκεύσει τις αλλαγές του.

/ Να συμπληρώσουμε ότι δεν είναι διαθέσιμη η επιλογή Delete, Edit και Create για όλους τους πίνακες λόγω του ότι παρατηρήσαμε ότι το μοντέλο του ΜVC απαιτούσε να υπάρχει πρωτεύον κλειδί στον πίνακα για να μπορέσει να εκτελέσει την αντιστοιχεί λειτουργεία. Εφόσον δεν υπήρχαν πρωτεύων κλειδιά σε ορισμένους πίνακες ο χρήστης δεν έχει την δυνατότητα να εκτελέσει τις πιο πάνω λειτουργίες. Στους πίνακες αυτούς αφαιρέθηκαν και τα κουμπιά για τις αντίστοιχες λειτουργείες.



Details.cshtml

Το αρχείο αυτό υπάρχει μόνο για τον πίνακα Publishers για την στήλη pub info. Έγινε η σελίδα αυτή λόγω του ότι οι πληροφορίες για κάθε Pub ήταν αρκετές και χαλούσαν την ομοιομορφία του πίνακα μας. Στην σελίδα εμφανίζεται και το Logo της κάθε pub.

9 **VIEWS**



Index.cshtml

Στο αρχείο αυτό βρίσκεται η σχεδίαση του πίνακα, δηλαδή ο τρόπος εμφάνισης και μορφοποίησης των δεδομένων μας στον πίνακα και εφόσον ο χρήστης επιλέξει τον πίνακα του εμφανίζεται η αντίστοιχη σελίδα με τα στοιχεία του πίνακα.

■ \Shared

Στον φάκελο αυτό υπάρχουν τα αρχεία .cshtml τα οποία είναι κοινά σε όλο το project.

Layout.cshtml

Στο αρχείο Layout.cshtml υπάρχει κώδικας html για την σχεδίαση του navigation bar, header και footer. Λόγω του ότι το αρχείο βρίσκετε στον φάκελο Shared η σχεδίαση θα εμφανίζεται σε όλες της σελίδες του project. Στην ουσία θα έχουμε κοινό navigation bar, header και footer σε κάθε σελίδα μας.

Global.asax.cs

Σε αυτό το αρχείο βρίσκεται ο κώδικας για την Error σελίδα μας. Αν ο χρήστης βρεθεί στην ανεπιθύμητη κατάσταση για όλους μας και πρέπει να αντικρίσει την σελίδα NOT FOUND, γίνεται Redirect το αρχείο NotFound.cshtml που βρίσκεται στον φάκελο Error.

\Error



Στο αρχείο αυτό βρίσκεται η custom error page μας. Σχεδιάσαμε την δική μας not found σελίδα για να είναι πιο κοντά στον χρήστη. Επίσης προσθέσαμε και ένα κουμπί για να μπορεί να επιστρέφει στην αρχική σελίδα όποτε αντικρίζει ένα not found page.

VIEWS 10

ANAZHTHΣΗ ΣΕ ΠΙΝΑΚΑ ΚΑΙ ΧΡΗΣΗ PAGED LIST

Η δημιουργία/λειτουργεία αναζήτησης και χρήσης paged list έγινε με την βοήθεια δύο packeges. Τα δύο αυτά πακέτα είναι το PageList.Μνς και PagedList.

Αναζήτηση:

Για παράδειγμα εδώ βλέπουμε στον Controller του Jobs ότι μέσα στο Index επιστρέφουμε ένα view (με δύο παραμέτρους, το string για τα αποτελέσματα της αναζήτησης του χρήστη και ένα int για το μέγεθος του paged list εφόσον κάθε φορά παραμετροποιείται ανάλογα με τον αριθμό των εγγραφών) όπου μέσα υπάρχει ένα query για την βάση δεδομένων μας. Το query ψάχνει αν υπάρχει διαθέσιμη εγγραφή ανάλογα με το τί θα αναζητήσει ο χρήστης για κάθε στήλη ταυτόχρονα και επιστρέφει σε μία λίστα τις διαθέσιμες εγγραφές αλλιώς αν δεν αναζητήσει κάτι ο χρήστης επιστρέφει όλες τις εγγραφές του πίνακα.

Paged List:

Το page list δημιουργείται δυναμικά σε κάθε πίνακα. Παρατηρήστε ότι όταν το View στο πιο πάνω παράδειγμα επιστρέφει μια λίστα (.toList()) καλείται και η μέθοδος .toPagedList() όπου μέσα στην μέθοδο επιστρέφουμε το pageNumber = 1 και το PageSize = 5, δηλαδή κάθε σελίδα θα επιστρέφει 5 εγγραφές από ολόκληρο τον πίνακα.

Αν έχουμε για παράδειγμα 24 εγγραφές θα δημιουργηθούν 5 σελίδες το pageList με 5 εγγραφές στις 4 σελίδες και 4 εγγραφές στην 5^η σελίδα. Αν προσθέσουμε ακόμα μία εγγραφή θα συμπληρωθεί η 5^η σελίδα πλήρως και αν δημιουργήσουμε επιπλέον ακόμα μία εγγραφή θα δημιουργηθεί μία 6^η σελίδα με μία εγγραφή μόνο.

PAGED LIST 11

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Σημειώσεις μαθήματος.
- https://stackoverflow.com
- https://forums.asp.net/1146.aspx
- https://www.youtube.com
- https://www.w3schools.com
- https://getbootstrap.com
- https://docs.microsoft.com/en-us/answers/products/sql-server
- https://docs.microsoft.com/en-us/sql/sqlserver/?redirectedfrom=MSDN&view=sql-server-ver15
- https://www.tutorialspoint.com/mvc framework/mvc framework int roduction.htm
- https://dotnet.microsoft.com/apps/aspnet/mvc

 $BIB\Lambda IO\Gamma PA\Phi IA$ 12

PACKAGES/EPFAAEIA

- Antlr
- Bootstrap.
- Bootstrap Icons.
- JQuery.
- JQuery.Validation.
- EntityFramework.
- FontAwesome.
- Microsoft.AspNet.Mvc
- Microsoft.AspNet.Razor
- Microsoft.AspNet.Web
- Microsoft.AspNet.Web.Optimization
- Microsoft.CodeDom.Providers.DotNetCompilerPlatform
- Microsoft.Data.SqlClient
- Microsoft.Data.SqlClient.SNI
- Microsoft.IdentityModel.JsonWebTokens
- Microsoft.IdentityModel.Logging
- Microsoft.IdentityModel.Protocols
- Microsoft.IdentityModel.Protocols.OpenIdConnect
- Microsoft.IdentityModel.Tokens
- Microsoft.jQuery.Unobtrusive.Validation
- Microsoft.Web.Infrastructure
- Modernizr
- Newtonsoft.Json
- popper.js
- System.Data.SqlClient
- System.IdentityModel.Tokens.Jwt
- WebGrease
- .NET Framework v.4.8.
- ReSharper.
- Git και Github

PACKAGES/EPTAAEIA 13