

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Σύστημα διαχείρισης περιεχομένου ιστού

WCMS HANDAX

Νικόλαος Χαμαλάκης

AM 1160

Επιβλέπων Καθηγητής : Ιωάννης Παχουλάκης

**ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΠΟΛΥΜΕΣΩΝ ΑΤΕΙ ΚΡΗΤΗΣ, 2008**

Σύστημα διαχείρισης περιεχομένου ιστού

WCMS HANDAX



Χαμαλάκης Νικόλαος
AM 1160

Περιεχόμενα

Εισαγωγή	4
Περιγραφή	4
Δυνατότητες	4
Αυτοματοποιημένα πρότυπα	4
Εύκολα τροποποιήσιμο περιεχόμενο.....	5
Εύκολη προσθήκη δυνατοτήτων.....	5
Βελτιώσεις προτύπων Web.....	5
Ροής διαχείρισης.....	5
Εικονικό περιεχόμενο	5
Τεχνολογίες	6
Ποιες τεχνολογίες εφαρμόστηκαν	6
Sql Server 2000.....	6
ASP NET 2.0	6
Γλώσσες .NET.....	6
To CLR (<i>Common Language Runtime</i>).....	6
The .NET Framework class library.....	6
Visual Studio 2005	7
Σχεδίαση σελίδων.....	7
Αυτόματη ανίχνευση λάθους.....	7
<i>Debugging tools</i>	7
Περιγραφή υλοποίησης	8
Ανάλυση βάσης δεδομένων	8
Πίνακες-Σχέσεις πινάκων.....	8
Πίνακας TABS.....	8
Πίνακας MODULES.....	8
Πίνακας MODULEDefs.....	9
Πίνακας HtmlText.....	9
Πίνακας Trees.....	9
Πίνακας ANNOUNCEMENTS.....	9
Πίνακας MODULESETTINGS.....	9
Πίνακας ROLES.....	10
Πίνακας USERS ROLES.....	10
Πίνακας USERS.....	11

Ανάλυση κυρίως προγράμματος.....	12
Αρχική σελίδα.....	12
URL rewriting.....	13
Modules.....	16
Module Actions.....	17
Action Print.....	17
Action Delete.....	17
HTML Module.....	18
HTML Edit Page.....	19
Announcements.....	20
Announcement σελίδα επεξεργασίας.....	21
Calendar.....	22
Clone Module.....	23
Clone Module Edit Page.....	23
Dynamic Module.....	24
Dynamic Module Edit Page.....	25
Page Edit.....	26
Tab Menu.....	27
Tree View.....	28
Login.....	29
Φάκελος admin.....	30
Access Denied.....	31
AddNewTab.....	32
Delete Module.....	32
Logoff.....	32
Module Settings.....	33
Tab Settings.....	34
Admin page.....	37
Site Settings.....	37
Security Roles.....	38
Tabs.....	38
Manage Users.....	39
Φάκελος Components.....	41
Κλάσεις.....	41
UsersDB.....	41
Announcements.....	41
DataCache.....	41
GlobalAsaxHelpe.....	42
HtmlTextDB.....	42
ModuleAction.....	42
TabHandler.....	42
Theme.....	42
Επίλογος.....	43

Εισαγωγή

Περιγραφή συστήματος διαχείρισης περιεχομένου ιστού

Ένα σύστημα διαχείρισης περιεχομένου Ιστού (WCMS) είναι ένα σύστημα διαχείρισης περιεχομένου (CMS) που υλοποιείται συνήθως ως εφαρμογή Ιστού (Web Application), για τη δημιουργία και τη διαχείριση αρχείων περιεχομένου HTML. Χρησιμοποιείται για να διαχειριστεί και να ελέγξει μια μεγάλη, δυναμική συλλογή του υλικού (έγγραφα HTML και τις συσχετιζόμενες με αυτά εικόνες). Ένα CMS διευκολύνει την δημιουργία, τον έλεγχο, την τροποποίηση του περιεχομένου αλλά υλοποιεί και πολλές ουσιαστικές λειτουργίες συντήρησης της εφαρμογής.

Επίσης το wcms επιτρέπει στο χρήστη που δεν έχει γνώσεις προγραμματισμού ή κάποιας script γλώσσας να μπορεί με εύκολο τρόπο να δημιουργήσει και να διαχειριστεί το περιεχόμενο.

Τα περισσότερα συστήματα χρησιμοποιούν μια βάση δεδομένων για να αποθηκεύσουν το περιεχόμενο, metadata, και οτιδήποτε άλλο μπορεί να απαιτεί το σύστημα.

Το περιεχόμενο συχνά, αλλά όχι πάντα μπορεί να αποθηκευτεί ως XML, για να διευκολύνει την επαναχρησιμοποίηση του και να επιτρέψει την εύκολη παρουσίαση του.

Η διαχείριση του συστήματος γίνεται μέσω του περιβάλλοντος επεξεργασίας από τον browser. μερικά συστήματα όμως απαιτούν άμεση πρόσβαση στο server όπου φιλοξενείται η εφαρμογή.

Δυνατότητες συστήματος διαχείρισης περιεχομένου ιστού

Μερικές από τις δυνατότητες που παρουσιάζει το WCMS αναφέρονται παρακάτω :

Αυτοματοποιημένα πρότυπα

Δημιουργεί τυποποιημένα templates (συνήθως HTML και XML) που μπορούν με ευκολία να εφαρμοστούν αυτόματα σε νέο ή σε υπάρχον περιεχόμενο, τα οποία μπορούν με δυναμικό τρόπο να αλλάξουν αμέσως την όψη αλλά και τις λειτουργίες του site.

Εύκολα τροποποιήσιμο περιεχόμενο

Αφού το περιεχόμενο είναι διαχωρισμένο από τον κεντρικό πυρήνα του site μπορεί εύκολα να τροποποιηθεί και να ανανεωθεί χάρη στη λειτουργία του WCMS να διαθέτει ενσωματωμένο WYSIWYG editor. Αυτή η λειτουργία είναι που το κάνει και πιο φιλικό προς τους χρήστες που δεν διαθέτουν ειδικευμένες γνώσεις μια που μπορούν αμέσως να δουν το πώς διαμορφώνετε το site.

Εύκολη προσθήκη δυνατοτήτων

Στα περισσότερα CMS μπορούν να εγκατασταθούν εύκολα διάφορα modules τα οποία μπορούν να επεκτείνουν τη λειτουργία του σε όποια σημεία επιθυμεί ο χρήστης.

Βελτιώσεις προτύπων Web

Κάθε CMS μπορεί να ενσωματώσει τις τελευταίες δυνατότητες και πρότυπα που υπάρχουν κάθε τεχνολογική περίοδο στο web ώστε να είναι πάντα επίκαιρο και να ανταποκρίνεται στις ανάγκες κάθε εποχής

Ροής διαχείρισης

Ροή διαχείρισης είναι ο κύκλος που χρειάζεται κάθε διαδικασία για να ολοκληρωθεί.

Παράδειγμα ένας χρήστης γράφει κάποιο άρθρο στην εφαρμογή αλλά δεν δημοσιεύετε στον ιστό αν δεν καθαριστεί ο κώδικας html που δημιουργείτε αλλά και μέχρι να το εγκρίνει ο υπεύθυνος περιεχομένου

Εικονικό περιεχόμενο

Τα συστήματα CMS μπορούν να παρέχουν μέσα σε κάθε χρήστη για να εργαστούν μέσα σε ένα εικονικό αντίγραφο του ολόκληρης της εφαρμογής, του συνόλου εγγράφων, ή και της βάσης δεδομένων. Αυτό επιτρέπει στις αλλαγές από πολλαπλά εξαρτώμενους πόρους να εμφανίζονται ανεξάρτητα και διαδοχικά .

Τεχνολογίες

Ποιες τεχνολογίες εφαρμόστηκαν

Το συγκεκριμένο WCMS για την αποθήκευση της πληροφορίας και του περιεχομένου των σελίδων αλλά και με ότι άλλο σχετίζεται με αυτές χρησιμοποιεί βάση δεδομένων. Για το λόγο αυτό έχει επιλεγεί ως βάση αποθήκευσης ο MS SQL SERVER 2000.

Επίσης για την υλοποίηση όλων των απαραίτητων λειτουργιών που είναι απαραίτητες για την δημιουργία και επεξεργασία του περιεχομένου στην βάση για το πώς αυτό θα εμφανίζεται αλλά και για πιο βαθιά ενσωματωμένες λειτουργίες ,έχει χρησιμοποιηθεί η αντικειμενοστραφής γλώσσα προγραμματισμού C# στην πλατφόρμα MS ASP NET 2.0.Τέλος η ανάπτυξη έχει γίνει στο περιβάλλον εργασίας MS Visual Studio 2005.

Sql Server 2000

Ο Microsoft SQL Server 2000 είναι ένα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων (RDBMS) που παράγεται από τη Microsoft. Η γλώσσα διατύπωσης ερωτημάτων της είναι η SQL..Μπορεί να φιλοξενήσει παραπάνω από μία βάσεις και να δέχεται ερωτήματα σε κάθε βάση από πολλαπλούς χρήστες .

ASP NET 2.0

Το ASP NET είναι η μηχανή η οποία φιλοξενεί τις web εφαρμογές και τις web υπηρεσίες. Με οποιοδήποτε χαρακτηριστικό γνώρισμα από τη βιβλιοθήκη .NET .Επίσης περιλαμβάνει και ένα σύνολο συγκεκριμένων υπηρεσιών web.

Το 2.0 υποδηλώνει ότι χρησιμοποιούμε το Framework 2.0.Το Framework 2,0 είναι ένα σύνολο τεχνολογιών :

Γλώσσες .NET :

Περιλαμβάνονται οι C # .NET , VB .NET , επίσης. JScript NET (μια δευτερεύουσα έκδοση της JavaScript), οι j # NET (αντιγραφή της JAVA), και το C ++ με τις ανάλογες επεκτάσεις .

Το CLR (Common Language Runtime):

Το CLR είναι η μηχανή που εκτελεί όλα. τα προγράμματα NET και παρέχει τις αυτόματες υπηρεσίες για αυτές τις εφαρμογές, όπως ο έλεγχος ασφάλειας, η διαχείριση μνήμης, και η βελτιστοποίηση.

The .NET Framework class library:

Είναι μια συλλογή από χιλιάδες κομμάτια τα οποία είναι προκατασκευασμένα και μπορούμε να τα κολλήσουμε στις εφαρμογές μας. Αυτά τα προκατασκευασμένα κομμάτια κώδικα οργανώνονται σε σύνολα εφαρμογών, όπως το ADO NET (βιβλιοθήκη για την δημιουργία των εφαρμογών βάσεων δεδομένων) και οι WINDOWS FORMS (η βιβλιοθήκη για τη δημιουργία των desktop εφαρμογών).

Visual Studio 2005

Το τελευταίο μέρος του .NET είναι το προαιρετικό εργαλείο ανάπτυξης visual studio, το οποίο παρέχει ένα πλούσιο περιβάλλον όπου μπορούμε γρήγορα να δημιουργήσετε μεγάλης κλίμακας εφαρμογές.

Μερικά από τα χαρακτηριστικά γνωρίσματα του visual studio, περιλαμβάνουν τα εξής

Σχεδίαση σελίδων:

Μπορείτε να δημιουργήσετε μια ελκυστική σελίδα με την ευκολία χρησιμοποιώντας το ενσωματωμένο Toolbox απλά τραβώντας τα components που θέλουμε στην σελίδα μας. Χωρίς να πρέπει να καταλάβετε τον κώδικα HTML.

Αυτόματη ανίχνευση λάθους:

Θα μπορούσατε να σπαταλάτε ώρες της εργασίας μέχρι να εντοπίσετε ένα λάθος στον κώδικα σας αλλά το visual studio ανιχνεύει και επισημαίνει το λάθος προτού καν να τρέξετε την εφαρμογή σας.

Debugging tools:

Το visual studio, διαθέτει πολύ καλά εργαλεία διόρθωσής τα οποία μας επιτρέπουν στην ροή του κώδικα να παρακολουθούμε τις μεταβλητές και πως αυτές ενεργούν. Εκτός από τις desktop εφαρμογές μπορούμε να ελέγξουμε και τις εφαρμογές web αφού διαθέτει ενσωματωμένο web server

Περιγραφή υλοποίησης

Ανάλυση βάσης δεδομένων

Όπως αναφέρθηκε παραπάνω το συγκεκριμένο WCMS για την αποθήκευση των δεδομένων χρησιμοποιεί σχεσιακή βάση δεδομένων. Η βασική ιδέα για την ανάπτυξη της βάσης ήταν ότι θα έπρεπε να είναι όσο το δυνατόν ευέλικτη με σαφή συσχέτιση των δεδομένων και την γρήγορη ανάκτηση ή τροποποίηση τους.

Επίσης το κύριο πρόγραμμα θα έπρεπε να αλληλεπιδρά με την βάση όσο το δυνατόν λιγότερο και όλες οι συναλλαγές να γίνονται γρήγορα και άμεσα ώστε να μην κουράζετε ο χρήστης.

Το περιεχόμενο που προβάλετε μέσω του WCMS παρουσιάζετε και ανανεώνετε δυναμικά. Αυτό σημαίνει ότι δεν υπάρχουν σελίδες με στατικό περιεχόμενο αλλά ανάλογα το ποια σελίδα θα ζητήσει ο χρήστης θα γίνει το κατάλληλο ερώτημα στη βάση δεδομένων θα επιλεγεί από τη βάση ότι σχετίζετε με αυτήν την σελίδα και μέσω αυτής της πληροφορίας θα δημιουργηθεί η σελίδα.

Πίνακες-Σχέσεις πινάκων:

Πίνακας TABS

Έτσι προκύπτει η ανάγκη να φτιάξουμε ένα πίνακα ο οποίος θα κρατάει τα χαρακτηριστικά της σελίδας ο οποίος θα είναι και ο κεντρικός πίνακας της βάσης. Έτσι εδώ αποθηκεύεται το όνομα σελίδας, ψευδώνυμο (χρησιμεύει στο Url rewriting θα αναλυθεί παρακάτω) ,ποιοι ρόλοι χρηστών θα μπορούν να επεξεργάζονται το περιεχόμενο της αλλά και ποιοι ρόλοι χρηστών θα έχουν πρόσβαση στη σελίδα, σε ποια θέση θα βρίσκεται η σελίδα ,αν έχει κάποια γονική σελίδα δηλαδή αν θα βρίσκεται κάτω από κάποια άλλη σελίδα στο menu και αν θα έχει κάποια πρότυπη σελίδα δηλαδή αν θα δανείζετε από κάποια άλλη σελίδα τα χαρακτηριστικά της και συστατικά της για να μην τα προσθέτουμε ξανά αφού ήδη υπάρχουν σε κάποια άλλη σελίδα.

Πίνακας MODULES

Πέρα από τα χαρακτηριστικά στοιχεία της σελίδας θα πρέπει να υπάρχει να υπάρχει και ένας πίνακας για να ορίζει και να αντιστοιχεί το περιεχόμενο της σελίδας. Επειδή υπάρχει ποικιλία περιεχομένου και πολύ διαφορετικοί τρόποι για να παρουσιαστεί το περιεχόμενο αυτός ο πίνακας κρατάει τα modules που αντιστοιχούν σε κάθε σελίδα. Τα modules είναι υποενότητες του προγράμματος ξεχωριστά ανεπτυγμένες αλλά και ανεξάρτητες μεταξύ τους. Σκοπός τους είναι μια συγκεκριμένη λειτουργία ως προς την προβολή του περιεχομένου που τους ανήκει ή ως προς την επεξεργασία του. Έτσι αυτός ο πίνακας που συνδέεται με τον πίνακα TABS (I-M) κρατάει το όνομα του module, τον τίτλο του,σε ποια θέση της σελίδας βρίσκεται ,ποιο ρόλοι έχουν δικαίωμα να το δούν να το επεξεργαστούν αλλά και αν έχει κάποιο περίγραμμα γύρο του (θέμα).

Πίνακας MODULEDefs

Όπως προαναφέρθηκε τα modules είναι ξεχωριστές ενότητες της εφαρμογής που υπάρχουν αυτόνομα. Έτσι δημιουργήθηκε αυτός ο πίνακας για κρατάει τα στοιχεία τους και να αναφέρετε σε αυτά. Εδώ λοιπόν αποθηκεύετε το κανονικό όνομα του module αλλά και σε ποιο μονοπάτι βρίσκεται μέσα στην εφαρμογή έτσι μπορεί να προσπελαθεί από το κύριο πρόγραμμα και να γίνει render στην σελίδα μας

Εκτός από τους βασικούς πίνακες υπάρχουν και οι πίνακες που αποθηκεύεται το περιεχόμενο κάθε module αυτοί οι πίνακες είναι:

Πίνακας HtmlText

Εδώ αποθηκεύεται το περιεχόμενο του HTMLMODULE. Το περιεχόμενο αυτό υπάρχει στη βάση σε μορφή HTML. Το πως γίνεται η αποθήκευση του εδώ και η ανάκτηση του θα αναλυθεί παρακάτω.

Πίνακας Trees

Σε αυτόν των πίνακα αποθηκεύετε το περιεχόμενο που προβάλλεται από modules Tree view. Τα στοιχεία που αποθηκεύονται σε αυτόν τον πίνακα ο τίτλος του συνδέσμου, σε ποιο μέρος θα μας πλοηγήσει αν είναι κύριος κόμβος ή όχι και αν μας ανοίξει νέο παράθυρο κατά την πλοήγηση μας

Πίνακας ANNOUNCEMENTS

Από το module ANNOUNCEMENTS το οποίο χρησιμοποιείται για την προβολή των ανακοινώσεων της εφαρμογής αποθηκεύεται εδώ το περιεχόμενο της κάθε ανακοίνωσης. Επίσης κρατείτε πληροφορία για το ποιος έγραψε την ανακοίνωση το πότε αλλά και το πότε σταματάει να ισχύει η κάθε ανακοίνωση .

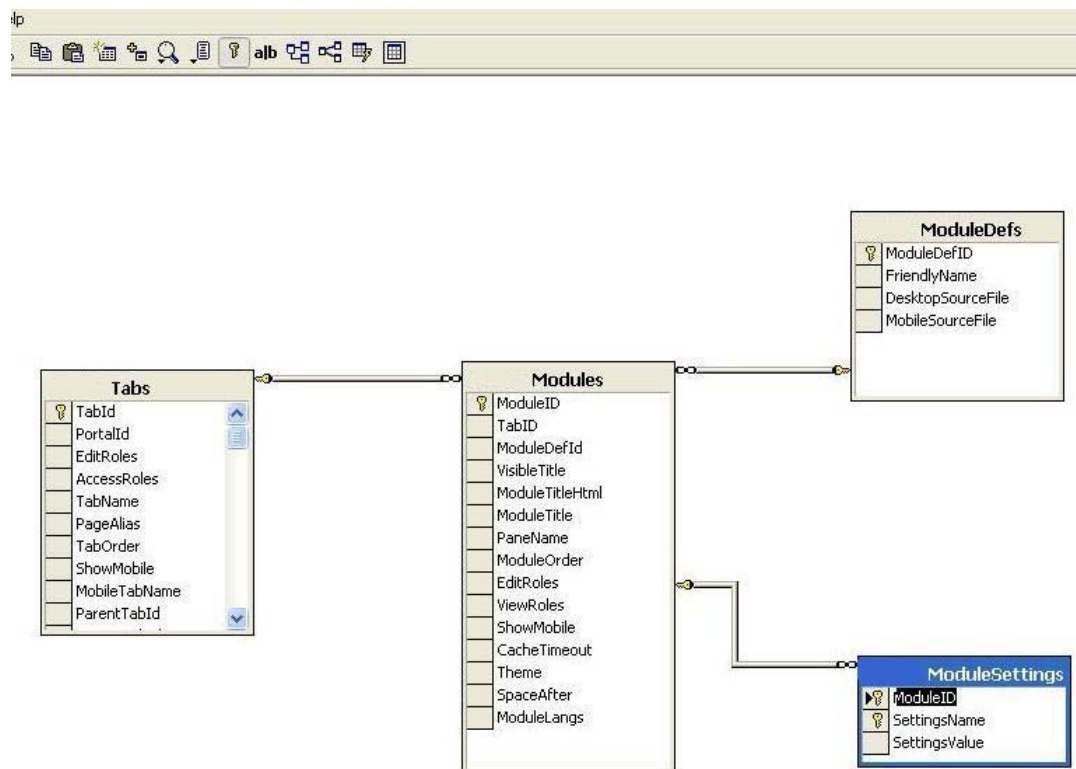
Πίνακας MODULESETTINGS

Πολλές είναι αναγκαίο να χρησιμοποιήσουμε το ίδιο περιεχόμενο σε παραπάνω από ένα μέρη στην ίδια σελίδα ή σε παραπάνω από μία σελίδες. Ποιο συγκεκριμένα να χρησιμοποιήσουμε το ίδιο module του προβάλλει αυτό το περιεχόμενο.

Για να αποφύγουμε αυτήν την επανάληψη έχουμε δημιουργήσει το Clone Module το οποίο αντιγράφει ένα ήδη υπάρχον module .Έτσι σε αυτόν τον πίνακα η πληροφορία που αποθηκεύεται μας λέει ότι το νέο clone module που δημιουργήθηκε θα αντιγράφει και θα έχει το ίδιο περιεχόμενο με το (X) module. Αυτό γίνεται καταχωρώντας στον πίνακα τα IDs των modules από τον πίνακα MODULES που παίρνουν κατά τη δημιουργία τους

Σύστημα διαχείρισης περιεχομένου ιστού

Οι πίνακες και οι μεταξύ τους σχέσεις φαίνονται στο παρακάτω διάγραμμα



Εκτός από τους πίνακες που κρατάνε το περιεχόμενο της εφαρμογής είναι και οι πίνακες που ορίζουν τα δικαιώματα κάθε χρήστη αλλά και αποθηκεύουν τα στοιχεία τους. Όπως είπαμε παραπάνω σε κάθε σελίδα αλλά και σε κάθε module που υπάρχει στην σελίδα μπορούμε να ορίσουμε ποιος θα βλέπει τι.

Έτσι προκύπτουν οι εξής πίνακες.

Πίνακας ROLES

Σε αυτόν τον πίνακα υπάρχουν αποθηκευμένοι οι ρόλοι των χρηστών. Τι δικαιώματα έχει ο κάθε ρόλος στην εφαρμογή μας και σε ποια σημεία έχει πρόσβαση ρυθμίζεται προγραμματιστικά από τον κώδικα της εφαρμογής.

Πίνακας USERS ROLES

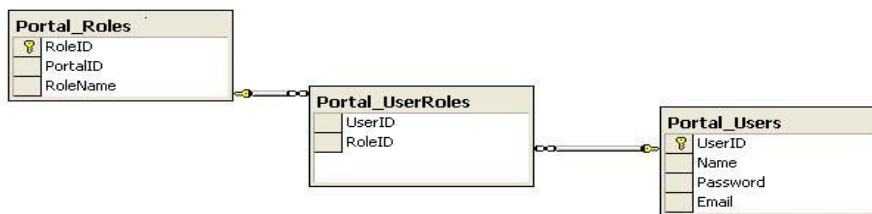
Εδώ αντιστοιχίζονται οι χρήστες με τους ρόλους.

Πίνακας USERS

Όλα τα στοιχεία του κάθε χρήστη που έχει μία παραπάνω δυνατότητα στην εφαρμογή πέρα από το να βλέπει το προεπιλεγμένο περιεχόμενο είναι αποθηκευμένα σε αυτόν τον πίνακα.

Όνομα, επίθετο ,EMAIL και ότι άλλο χαρακτηριστικό από τον χρήστη επιθυμούμε μπορεί να αποθηκευτεί εδώ. Αυτά είναι και τα στοιχεία με το οποία γίνεται η είσοδος η επικύρωση και η έγκριση της πρόσβασης τους στου πόρους της εφαρμογής .

Οι πίνακες και οι μεταξύ τους σχέσεις φαίνονται στο παρακάτω διάγραμμα



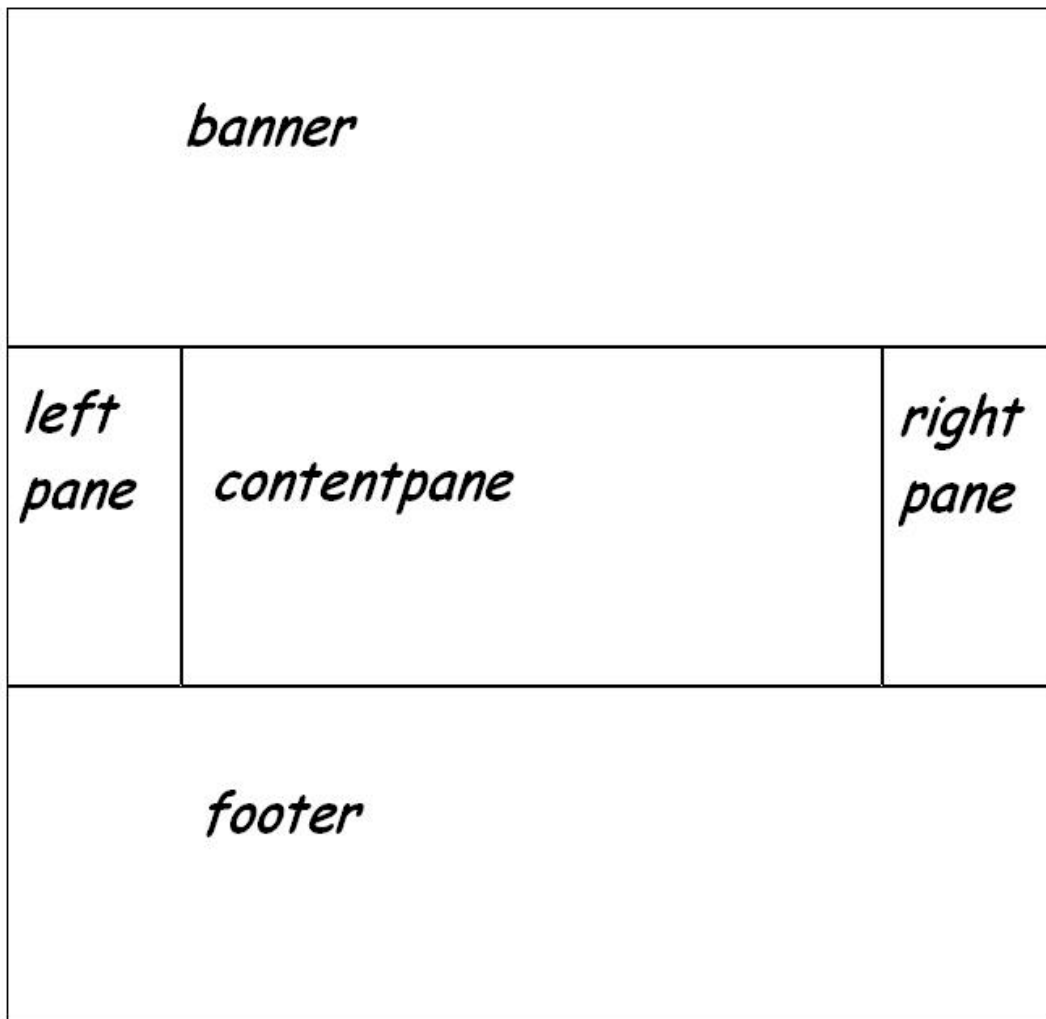
Ανάλυση κυρίως προγράμματος

Όπως έχουμε πει όλο το περιεχόμενο της εφαρμογής βρίσκεται αποθηκευμένο μέσα στην βάση δεδομένων, έτσι δεν χρειάζεται να έχουμε σελίδες με στατικό περιεχόμενο στην εφαρμογή.

Επίσης έχει αναφερθεί ότι το περιεχόμενο εμφανίζεται μέσα από τα modules που τοποθετούνται στην σελίδα. Έτσι έχουμε μια σελίδα κενή της οποίας το περιεχόμενο ανανεώνεται κάθε φορά ανάλογα το Url που έρχεται, μέσα στο οποίο είναι ορισμένη η σελίδα επιθυμούμε και σύμφωνα με αυτό επιλέγεται το περιεχόμενο που θα προβληθεί.

Θυμίζουμε ότι στο module έχουμε ορίσει σε ποια θέση τοποθετείται στην σελίδα αλλά και σε ποια σελίδα ανήκει. Για την κάθε σελίδα δεν φτιάχνεται νέο module αλλά ένα module φορτώνεται σε κάθε σελίδα με διαφορετικό περιεχόμενο που ορίζεται κάθε φορά από την σελίδα που φορτώνεται.

Αρχική σελίδα



Στο παραπάνω σχήμα φαίνεται η μορφή της κεντρικής σελίδας και το πώς είναι χωρισμένη σε τμήματα στα οποία θα τοποθετηθούν τα διάφορα modules. Η σχεδίαση της είναι σταθερή και δεν αλλάζει ανάλογα με το ποια σελίδα το περιεχόμενο θα εμφανίζεται. Σε κάθε τμήμα της μπορούν να τοποθετηθούν παραπάνω από ένα module ίδιου ή διαφορετικού τύπου.

Το όνομα της σελίδας είναι Default.aspx και είναι ορισμένη σαν αρχική σελίδα στο web application που υλοποιήσαμε. Έτσι όποτε πλοηγούμαστε στην εφαρμογή μας η πρώτη σελίδα που θα εμφανίζεται θα είναι η default.aspx μόνο που θα είναι κενή. Το τι περιεχόμενο θα εμφανιστεί στη σελίδα όπως έχουμε προαναφέρει εξαρτάται από το url ποια σελίδα θα ζητείται.

Σε αυτό το σημείο θα πρέπει να αναλύσουμε τα περιεχόμενα του Url αλλά και το url rewriting που εκτελείται πριν από την δημιουργία της σελίδας.

URL rewriting

Το url που δέχεται η εφαρμογή μας είναι της εξής μορφής. :

http://localhost/Handax/Default.aspx

Το συγκεκριμένο url είναι το φυσικό μονοπάτι της εφαρμογής το οποίο μας εμφανίζει μια κενή σελίδα. Δεν μπορούμε να ορίσουμε μια σελίδα να εμφανίζεται κάθε φορά που ανοίγει η εφαρμογή μιας και δεν υπάρχει στατικό περιεχόμενο σε κάποια σελίδα αφού οι σελίδες υπάρχουν εικονικά και είναι αποθηκευμένες στην βάση δεδομένων μας. Όμως ούτε από την βάση μπορούμε με στατικό τρόπο να τραβάμε ένα TABID και αντίστοιχα τα περιεχόμενα της σελίδας που ανήκουν σε αυτό το TABID αφού τροποποιώντας το περιεχόμενο της εφαρμογής είναι πολύ πιθανό να διαγραφεί και να εμφανιστεί πάλι κενή σελίδα ζητώντας αυτό το TABID.

Λαμβάνοντας υπ όψιν μας ότι μερικές σελίδες δεν είναι προσβάσιμες για μερικούς ρόλους χρηστών δεν θα μπορούσαμε να τραβάμε τυχαία κάποιο TABID σελίδας και να εμφανίζουμε αυτό το περιεχόμενο γιατί θα αντιμετωπίζαμε πρόβλημα πρόσβασης στο συγκεκριμένο περιεχόμενο. Επίσης ο διαχειριστής της εφαρμογής έχει την δυνατότητα να δίνει κατά βούληση στις σελίδες όποιο όνομα θέλει. Με αυτήν την ελευθερία ίσως να μην ονομάσει κάποια σελίδα HomePage ή Home ή Αρχική για να καταφέρουμε από το όνομα της να εντοπίσουμε την αρχική σελίδα και να την εμφανίσουμε στο χρήστη.

Για τους παραπάνω λόγους λοιπόν αποφασίσαμε η σελίδα που θα εμφανίζεται στον χρήστη αν δεν περιέχει το url κάποιο TABID σελίδας να είναι η μικρότερη σε νούμερο TABID που έχουν δικαίωμα όλοι οι χρήστες να προσπελάσουν. Όποτε το προηγούμενο url μετατρέπεται σε :

http://localhost/Handax/Default.aspx ? Tabid=1

Το tabid=1 φανερώνει ότι θα εμφανιστεί το περιεχόμενο της σελίδας που είναι αποθηκευμένη στην βάση δεδομένων με TABID =1 ,στη δική μας εφαρμογή τυγχάνει να είναι και η αρχική μας σελίδα .

Η κανονική λειτουργία και χρήση του Url rewrite δεν είναι αυτή που μόλις αναφέραμε αλλά το πώς ο χρήστης αλληλεπιδρά με τις διευθύνσεις για την δική του ευκολία.

Είναι γνωστό ότι οι χρήστες των εφαρμογών προτιμούν μικρά, τακτοποιημένο URLs από τις τερατώδεις διευθύνσεις που συναντάμε συνήθως οι οποίες είναι δύσκολο να κατανοηθούν από τις παραμέτρους που περιέχουν . Κατά

Σύστημα διαχείρισης περιεχομένου ιστού

διαστήματα, ο χρήστης θα πρέπει να είσαι σε θέση να θυμηθεί και να πληκτρολογήσει ένα μικρό URL όπου να είναι λιγότερο χρονοβόρο από να προσθέσει την σελίδα στις αγαπημένες τοποθεσίες του Browser και να πρέπει να την ξαναψάξει από κει . Ας συγκρίνουμε τις παρακάτω διευθύνσεις
[Http://www.somebloghost.com/Blogs/Posts.aspx?Year=2006&Month=12&Day=10](http://www.somebloghost.com/Blogs/Posts.aspx?Year=2006&Month=12&Day=10)
Και (2) <http://www.somebloghost.com/Blogs/2006/12/10/>

Το πρώτο URL περιέχει τις παραμέτρους για να κωδικοποιήσει την ημερομηνία για την οποία μερικές μηχανές blog πρέπει να παρουσιάσουν διαθέσιμες ταχυδρομήσεις. Το δεύτερο URL περιέχει αυτές τις πληροφορίες στη διεύθυνση, που δίνει στο χρήστη μια σαφή ιδέα για το τι πρόκειται να δει. Η δεύτερη διεύθυνση επιτρέπει επίσης στο χρήστη για να τροποποιήσει το URL για να δει όλες τις ταχυδρομήσεις διαθέσιμες το Δεκέμβριο, απλά με την αφαίρεση του κειμένου που κωδικοποιεί την ημέρα "10": <http://www.somehost.com/Blogs/2006/12/>.

Ένας άλλος λόγος πέρα από την ευχρηστία που μόλις αναφέραμε είναι η ανακατασκευή των Url.

Στις μεγάλες εφαρμογές είναι συνηθισμένο για τους υπεύθυνους ανάπτυξης να κινούν τις σελίδες από τον έναν κατάλογο προς τον άλλο. Ας υποθέσουμε ότι οι σελίδες αρχικά ήταν διαθέσιμες στο Url

<http://www.somebloghost.com/Info/Copyright.aspx> και
<http://www.somebloghost.com/Support/Contacts.aspx>,

αλλά κάποια στιγμή οι υπεύθυνοι για δικούς τους λόγους μετακίνησαν το [Copyright.aspx](#) και την σελίδα [Contacts.aspx](#) σε έναν νέο φάκελο. Οι χρήστες που δεν γνωρίζουν για αυτή την αλλαγή θα πρέπει να ενημερωθούν και να οδηγηθούν στη νέα θέση. Αυτό το ζήτημα μπορεί να επιλυθεί με την προσθήκη απλών πλαστών σελίδων στις παλιές θέσεις οι οποίες να κάνουν redirect του χρήστες στην νέα τοποθεσία. Όμως εάν υπάρχουν εκατοντάδες μετακινημένων σελίδων σε όλη την εφαρμογή το πρόγραμμα θα περιέχει σύντομα πάρα πολλές άχρηστες σελίδες που θα έχουν μόνο τους σκοπό να οδηγούν τους χρήστες στη νέα θέση. Με το Url rewriting , μπορούμε να μεταβάλουμε το κανονικό μονοπάτι σε εικονικό μέσα από το configuration file της εφαρμογής και να όλα τα προβλήματα από την μετακίνηση των σε

Σύμφωνα με τα παραπάνω ο χρήστης αν είχε προχωρήσει την πλοήγηση του στην εφαρμογή και επιθυμούσε να επιστρέψει στην αρχική σελίδα θα μπορούσε πολύ απλά να πληκτρολογήσει

<http://localhost/Handax/HomePage.aspx> το οποίο είναι ίδιο με το **<http://localhost/Handax/Default.aspx>** ? Tabid=1 και μας οδηγούν στην αρχική σελίδα της εφαρμογής.

Αφού πρώτα αναφερθήκαμε γενικά στο Url rewrite εξηγήσαμε πως προέκυψε η ανάγκη του και πως γίνεται το rewrite από το χρήστη στο πρόγραμμα αλλά και από το πρόγραμμα στο χρήστη, θα αναλύσουμε τα δικά μας βήματα αλλά και τον τρόπο που έχουμε αναπτύξει αυτή την λειτουργία.

Η διαδικασία του rewrite γίνεται πριν καν φτάσει η εκτέλεση του προγράμματος στην αρχική σελίδα . Το url rewrite εκτελείται στο Application Begin Request μέσα στο αρχείο Global .asax .Στο σημείο αυτό δεν έχει γίνει ταυτοποίηση του χρήστη, οπότε αν κάποιος κακόβουλος πηγαινε να προσπελάσει μια σελίδα που δεν έχει δικαίωμα δεν θα του επιτρεπόταν η είσοδος.

Υποθέτουμε ότι ο χρήστης ζητάει να εκτελεστεί το εξής url **<http://localhost/Handax/HomePage.aspx>**, έτσι θα πρέπει να ψάξει η εφαρμογή για κάποια στατική σελίδα homepage.aspx και να την προβάλλει στο χρήστη. Λογικό επόμενο αφού όλο το περιεχόμενο είναι αποθηκευμένο στην βάση να μην υπάρχει η σελίδα homepage.aspx και να παρουσιαστεί σφάλμα στο server ότι δεν βρέθηκε η ζητούμενη σελίδα. Δική μας δουλειά είναι να εντοπίσουμε το όνομα της σελίδας και να ανακτήσουμε από την βάση το TABID της και να το μετατρέψουμε σε **[http://localhost/Handax/Default.aspx ? Tabid=\(x\)](http://localhost/Handax/Default.aspx?Tabid=(x))** αλλιώς και δεν υπάρχει αντίστοιχη σελίδα πάλι θα εμφανίζεται το μήνυμα λάθους.

Ας τα πάρουμε από την αρχή. Με το που αρχίσει να εκτελείτε η Begin Request πρώτο μας μέλημα είναι να δούμε αν πρέπει να κάνουμε το url rewrite. Οι περιπτώσεις που δεν εκτελούμε το rewrite είναι όταν η πλοήγηση γίνεται με το κανονικό μονοπάτι δηλαδή περιέχει το default.aspx και όχι κάποιο ψευδώνυμο σελίδας ή όταν ζητάμε μία σελίδα επεξεργασίας των ρυθμίσεων της εφαρμογής. Σε περίπτωση που πρέπει να εκτελεστεί το rewrite ελέγχουμε αν το μονοπάτι που δηλώνει τη θέση της σελίδας είναι έγκυρο. Δηλαδή αν ζητάμε κάποια σελίδα που είναι «παιδί» κάποιας άλλης θα πρέπει να επιβεβαιώσουμε ότι η διαδρομή για να φτάσουμε στην ζητούμενη σελίδα είναι όντος αυτή.

Η μεθοδολογία που εφαρμόσαμε είναι να περνούμε ένα ένα τα κομμάτια από το μονοπάτι και να κοιτάμε αν το ένα έχει το άλλο ως «παιδί» μέχρι να φτάσουμε και στο τελευταίο τμήμα της διαδρομής όπου και τελειώνει ο έλεγχος. Αν όλα πάνε καλά και επαληθευτεί το μονοπάτι τότε συνεχίζουμε την διαδικασία του rewrite αλλιώς και πάλι εμφανίζεται το λάθος 404 ότι δεν υπάρχει η ζητούμενη σελίδα. Το επόμενο βήμα είναι με το κατάλληλο sql ερώτημα να πάρουμε από την βάση δεδομένων το TABID που αντιστοιχεί στην σελίδα και να το προσαρτήσουμε στο «νέο» url. Έπιτα από όλους αυτούς τους ελέγχους και αφού έχουμε φέρει το url στην μορφή που επιθυμούμε με την εντολή **HttpContext.Current.RewritePath()**; γίνεται το rewrite και προχωράει η ροή του προγράμματος.

Τώρα η ροή του προγράμματος μας μεταφέρει στον κώδικα της σελίδας default.aspx. Επειδή η default.aspx είναι η κεντρική σελίδα και σε αυτή φορτώνονται όλες οι άλλες εικονικές σελίδες ο πρώτος έλεγχος που γίνεται είναι να ελέγξουμε αν ο τρέχον χρήστης έχει τα απαραίτητα δικαιώματα για να προσπελάσει την σελίδα και το περιεχόμενο της. Αν περάσει το έλεγχο ο χρήστης ξεκινάμε να φορτώνουμε τα module και το περιεχόμενο τους αλλιώς θα εμφανιστεί το μήνυμα ότι δεν έχει δικαίωμα πρόσβασης. Ο δεύτερος έλεγχος που γίνεται είναι αν ο χρήστης έχει δικαίωμα επεξεργασίας της σελίδας. Εφόσον έχει δικαίωμα ο χρήστης να επεξεργαστεί την σελίδα τότε προστίθεται το module επεξεργασίας της σελίδας.

Τις λειτουργίες αυτού του module αλλά και τον τρόπο που προστίθενται τα module στην σελίδα θα τους δούμε παρακάτω.

Μόλις περάσουμε αυτούς τους ελέγχους και ενώ έχουμε μάθει το TABID από το url επικοινωνούμε με την βάση για να μάθουμε ποια Modules ανήκουν σε αυτή την σελίδα αλλά και αν έχει κάποια κύρια σελίδα Master page. Αυτό γίνεται εκτελώντας ένα sql ερώτημα τα αποτελέσματα του οποίου τα κρατάμε σε ένα data Table. Τα δεδομένα που κρατάμε σε αυτό το πίνακα είναι από τον πίνακα Modules όλα τα στοιχεία του module όνομα, ID, κτλ από τον πίνακα moduledefs που βρίσκεται αποθηκευμένο στην εφαρμογή για να το βρούμε και να το τοποθετήσουμε στην σελίδα μας.

Η Master page για μας δεν είναι μια σελίδα ακριβώς αλλά ένα σύνολο από κάποια Module τα οποία είναι κοινά σε περισσότερες από μία σελίδες. Για να μην προσθέτουμε λοιπόν σε παραπάνω από μια σελίδες τα ίδια και τα ίδια Module με αποτέλεσμα να υπάρχει πλεονασμός και να φορτώνουμε την βάση μας χωρίς λόγο τα ενοποιούμε σε μια Master Page. Η master Page τα modules της πιο συγκεκριμένα τοποθετούνται στην σελίδα πριν το περιεχόμενο που της σελίδας που έχει ζητηθεί από τον χρήστη με συνέπεια όσες σελίδες έχουν κοινή master page να έχουν κοινή διάταξη. Αυτό μας βοηθάει σε κάποιες εφαρμογές που θέλουμε όλη η εφαρμογή να είναι ομοιογενής. Δημιουργώντας έτσι αυτή την σελίδα που προσαρμόζεται σε οποιαδήποτε άλλη αυτή η απαίτηση λύνεται πολύ εύκολα.

Πριν κάνουμε render το module στην σελίδα ελέγχουμε αν έχει το module κάποιο θέμα δηλαδή κάποιο διακοσμητικό περίγραμμα. Εάν έχει το module θέμα τότε πρώτα βάζουμε το θέμα στην σελίδα και μετά σε ειδική θέση μέσα στο θέμα προσθέτουμε το module, έτσι περιβάλλετε από αυτό το διακοσμητικό περίγραμμα. Επειδή το θέμα είναι ένα user control τοποθετείται στην σελίδα όπως τα module και μέσα σε αυτό το control τοποθετείται module. Δηλαδή έχουμε user control μέσα σε user control και μετά γίνονται render στην σελίδα. Στην περίπτωση που δεν έχει το module κάποιο θέμα τότε τοποθετείται στην σελίδα και μετά από το τμήμα του προγράμματος που εκτελείται μέσα στο Module που θα αναλύσουμε παρακάτω γεμίζει το module με το περιεχόμενο του.

Συνήθως στο Banner και στο Footer τοποθετούνται σήματα logos επωνυμίες οι οποίες θέλουμε να είναι χωρίς κενά μονοκόμματα. Ο τρόπος που εμφανίζουμε εικόνες η κείμενο είναι το Html Module. Στα module όμως για περισσότερη ευκολία και ευελιξία έχουμε κάποιες ενέργειες που αναφέρονται σε αυτά όπως εκτύπωση η επεξεργασία. Αν θέλουμε να εμφανίσουμε μια εικόνα η ένα σήμα θα πρέπει να το τοποθετήσουμε σε παραπάνω από ένα html Module που να συνδέονται μεταξύ τους. Αν όμως παραμένουν οι ενέργειες το module οι οποίες δηλώνονται με κάποια εικόνα διασπάται η συνέχεια του σήματος με αποτέλεσμα να χαλάνε τα γραφικά. Έτσι τα modules του banner και του footer φορτώνουμε ανεξάρτητα από τα άλλα τρία τμήματα της σελίδας για κάνουμε τον απαραίτητο έλεγχο ώστε να μην τοποθετούνται οι ενέργειες του κάθε module.

Παρακάτω θα προσπαθήσουμε να εξηγήσουμε τον τρόπο με τον οποίο έχουν δημιουργηθεί τα modules από ποια κλάση κληρονομούν τις ενέργειες τους, πως γίνεται η επεξεργασία τους, πως λειτουργούν αλλά και το πώς μας βοηθάνε και που ακριβώς τα χρησιμοποιούμε.

Modules

Τα modules είναι ανεξάρτητα κομμάτια της εφαρμογής τα οποία υπάρχουν μέσα της και εκτελούν το καθένα μια συγκεκριμένη δουλειά. Το ένα module από το άλλο είναι ανεξάρτητο αλλά μπορούν να συνυπάρχουν στην ίδια σελίδα και στο ίδιο τμήμα της σελίδας. Αν είναι απαραίτητο βέβαια παραπάνω από ένα module μπορούν να συνδυαστούν για να προβληθεί από κοινού κάποιο περιεχόμενο.

Τα modules δεν είναι τίποτα άλλο από user control. Τα user control είναι μικροεφαρμογές που μπορούν να υλοποιηθούν με το visual studio και το .NET. Πολλές φορές σε περισσότερα από ένα μέρη στην εφαρμογή μας χρειάζεται να εκτελέσουμε την ίδια εργασία. Έτσι για να αποφύγουμε πάλι τον πλεονασμό και το παραπάνω φόρτωμα της εφαρμογής εκτελούμε αυτήν την εργασία μέσα σε ένα module. Το module μπορεί να προστεθεί σαν control σε μια web σελίδα η σε οτιδήποτε μπορεί να σταθεί και να προβληθεί μόνο του από τον σελιδοδείκτη.

Εμείς όμως επειδή έχουμε παραπάνω από ένα module και επειδή θέλουμε μια ενιαία δομή για να τα διαχειριστούμε και επειδή μας χρειάζονται παραπάνω πληροφορίες για κάθε module περά από τα στοιχεία του user control δημιουργήσαμε μια νέα κλάση. Την κλάση αυτή την έχουμε ονομάσει Module Control η οποία κληρονομεί από την κλάση του συστήματος User Control και το κάθε module με την σειρά του κληρονομεί από την Module Control. Με την κλάση αυτή μπορούμε να δημιουργήσουμε αντικείμενα user control αλλά με ποιο πολλές μεταβλητές στις οποίες μπορούμε να κρατήσουμε πληροφορίες χρήσιμες για μας.

Έτσι στην κλάση αυτή κρατάμε το Module Id από τον πίνακα Modules που βρίσκεται στην βάση μας αλλά επίσης εκτελείται και μια συνάρτηση η οποία μας επιστρέφει όλες τις απαραίτητες πληροφορίες για αυτό το module τις οποίες αντλεί πάλι από τον πίνακα modules.

Τα modules τα οποία έχουμε υλοποιήσει και συμπεριλάβει στην εφαρμογή μας είναι τα έξι: Html module, Announcements , Dynamic module, calendar, Clone module, Login. Tab Menu, Tree View. Για κάθε module υπάρχει και μία επιπλέον σελίδα. Αυτή η σελίδα μας χρησιμεύει στην επεξεργασία των ιδιοτήτων και των ρυθμίσεων του module.

Επίσης κάθε module έχει και κάποιες ενέργειες που κολάνε πάνω του. Αυτές οι ενέργειες μπορεί να είναι εκτύπωση, επεξεργασία η και όποια άλλη κρίνουμε σκόπιμη ότι χρειάζεται παράδειγμα την αποστολή με mail του περιεχομένου του module. Έτσι και για κάθε ενεργεία υπάρχει και μια σελίδα όπου γίνεται αυτή η επεξεργασία και ενέργεια.

Module Actions

Οι ενέργειες του κάθε module υπάρχουν καταχωρημένες σε ένα αρχείο xml μέσα στο φάκελο του στην εφαρμογή. Την στιγμή που πρέπει να τοποθετηθεί το module στη σελίδα οι ενέργειες του μάλλον οι εικόνες που δηλώνουν την κάθε ενέργεια τοποθετούνται είτε στο θέμα που περιβάλλει το module είτε σε ειδική θέση μέσα στο χώρο του module. Πληροφορίες για το ποια εικόνα χρησιμοποιείται για κάθε σε κάθε ενέργεια το πώς ονομάζεται βρίσκονται μέσα στο xml οπότε διαβάζουμε το xml και αναλόγως βάζουμε και την κατάλληλη ενέργεια.

Επειδή οι ενέργειες της επεξεργασίας είναι διαφορετικές και φτιαγμένες ανάλογα τις απαιτήσεις του module θα τις αναλύσουμε ξεχωριστά σε κάθε module. Εδώ θα εξηγήσουμε μόνο την διαδικασία της εκτύπωσης και της διαγραφής που γίνονται με τον ίδιο τρόπο σε όλα τα modules.

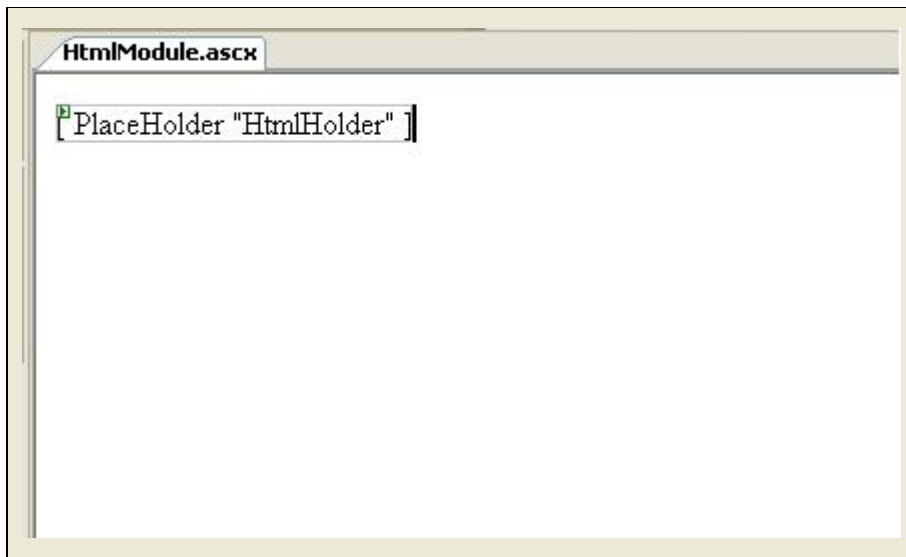
Action Print

Για την εκτύπωση του περιεχομένου του module έχουμε δημιουργήσει μια νέα σελίδα. Μόλις πατήσει ο χρήστης το κουμπί της εκτύπωσης γίνεται redirect στην σελίδα print.aspx. Σε αυτή την σελίδα φορτώνεται το περιεχόμενο του module που δημιούργησε αυτή την ενέργεια και με ένα java script κώδικα που εκτελείται στο body on load αυτής της σελίδας τυπώνεται και το ζητούμενο περιεχόμενο

Action Delete

Με την εντολή Delete διαγράφουμε από την βάση και από όλους τους πίνακες οτιδήποτε σχετίζεται με το συγκεκριμένο module.

HTML Module



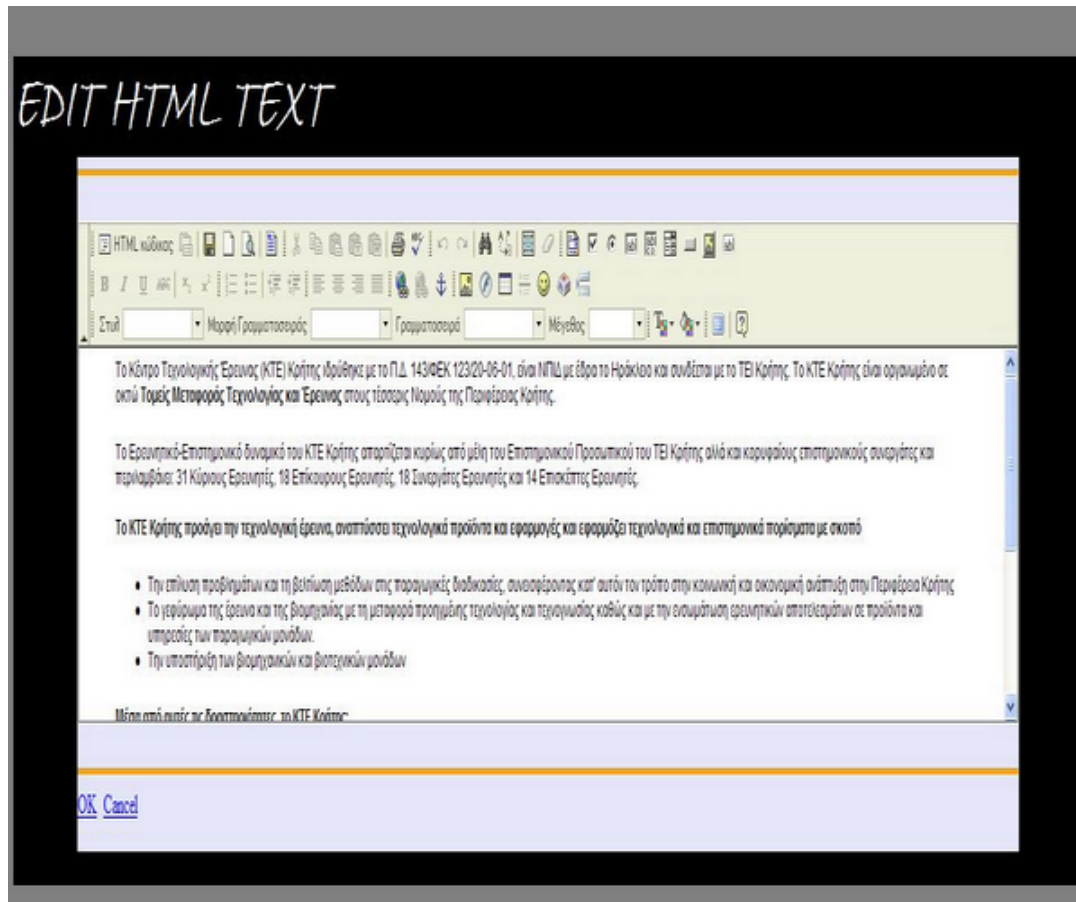
Το Html module όπως μπορούμε να δούμε και στην εικόνα δεν είναι τίποτα άλλο από ένα user control το οποίο έχει μέσα του ένα Placeholder. Ο place holder είναι ένα control του visual studio στο οποίο μπορούμε να προσθέσουμε κάθε λογής περιεχόμενο. Το html module μας δίνει την δυνατότητα να εμφανίσουμε περιεχόμενο html δηλαδή ότι μπορούμε να γράψουμε σε ένα κειμενογράφο html και μετά να εμφανιστεί σε κάποιον σελιδοδείκτη μπορούμε να το παρουσιάσουμε μέσα από αυτό το module.

Μετά που θα γίνει render το Module στην σελίδα, αρχίζει και εκτελείται ο κώδικας που βρίσκεται μέσα του. Έχουμε αναφέρει ότι στην βάση υπάρχει ένας πίνακας που ονομάζεται Html Text σε αυτόν τον πίνακα υπάρχουν αποθηκευμένα το περιεχόμενο για κάθε html module με το ID τους. Η μορφή που είναι αποθηκευμένο το περιεχόμενο του module είναι σε encoded html και ο λόγος είναι για να επαλειφθούν οι μη έγκυροι χαρακτήρες html.

Ξέρουμε το Module Id αφού όπως είπαμε το κρατάμε στις μεταβλητές του αντικειμένου module Control από όπου και κληρονομεί. Με αυτό το στοιχείο παίρνουμε από την βάση το περιεχόμενο που επιθυμούμε. Μόλις έχουμε ανακτήσει από την βάση το περιεχόμενο το κάνουμε decode και το προσθέτουμε στο περιεχόμενο του Placeholder και είμαστε έτοιμοι να το προβάλλουμε στον χρήστη.

Εάν θέλουμε να επεξεργαστούμε το περιεχόμενο που προβάλλεται μέσα από αυτό το module θα πρέπει να πατήσουμε την ενέργεια Edit και να μας κάνει redirect στην σελίδα επεξεργασίας του Html module.

HTML Edit Page

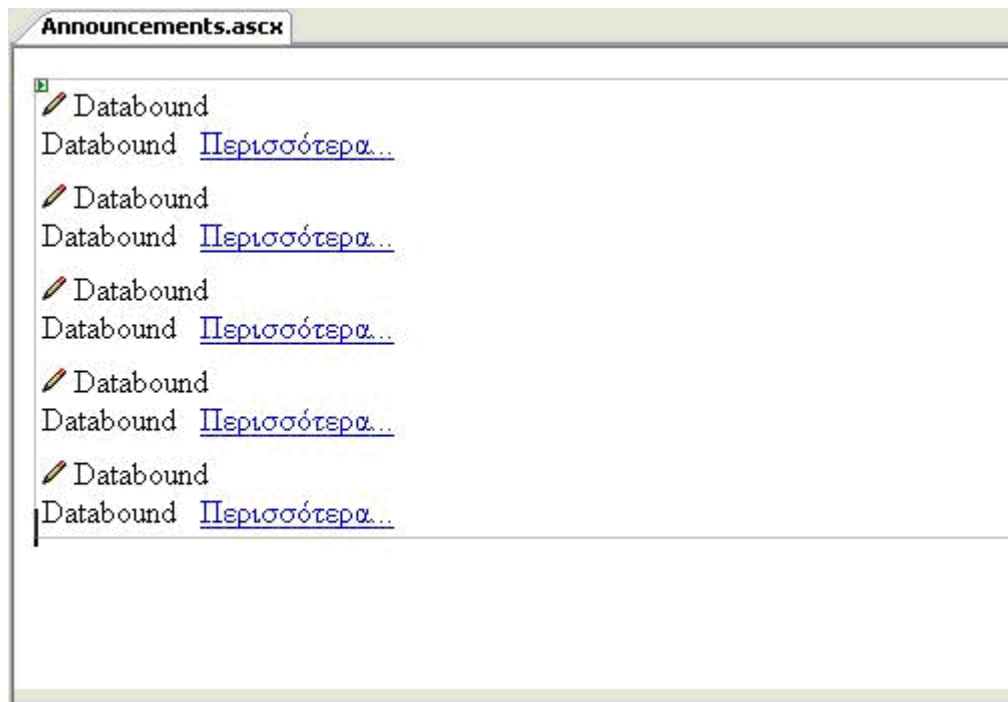


Αυτή είναι η σελίδα επεξεργασίας του Html module. Για την επεξεργασία του περιεχομένου έχουμε βάλει ένα html editor τον Fck editor συγκεκριμένα (<http://www.FckEditor.com>) ο οποίος είναι freeware. Το περιεχόμενο του module εμφανίζεται μέσα στον editor ακριβώς με τον ίδιο τρόπο όπως και πριν στον place holder.

Η διαφορά μας εδώ είναι ότι μπορούμε να επεξεργαστούμε το περιεχόμενο όπως επιθυμούμε σαν έναν κανονικό κειμενογράφο, να αλλάξουμε γραμματοσειρές να προσθέσουμε εικόνες κτλ. Μετά το πέρας της επεξεργασίας το περιεχόμενο θα πρέπει να αποθηκευτεί ξανά πίσω στην βάση. Πριν γίνει η ενημέρωση της βάσης το περιεχόμενο ξαναγίνεται encoding και μετά αποθηκεύεται.

Το Html module είναι το πιο βασικό Module για την παρουσίαση του περιεχομένου αφού μέσα από αυτό μπορούμε να παρουσιάσουμε τα πάντα με όποιον τρόπο θέλουμε και να δώσουμε την μορφή που επιθυμούμε στην εφαρμογή μας. Το βασικό είναι ότι όλα αυτά μπορούν να γίνουν πολύ εύκολα αφού υπάρχει ο κειμενογράφος οι δυνατότητες του οποίου είναι υπεραρκετές και μπορούν να καλύψουν όλες τις απαιτήσεις

Announcements



Το module αυτό είναι το δεύτερο κατά σειρά module που θα επεξηγήσουμε. Από το όνομα του καταλαβαίνουμε ότι θα αναρτούμε κάποιες ανακοινώσεις ή απλά πληροφορίες που θα χρειάζονται λίγο παραπάνω εξήγηση ή θα πρέπει να μην φαίνονται μετά από κάποιο χρονικό διάστημα.

Όπως βλέπουμε στον πυρήνα του το module αυτό έχει μια data list. Η data list είναι ένα control του visual studio το οποίο μπορεί να επαναλαμβάνει παρόμοια δεδομένα από μία πηγή δεδομένων. Επίσης μέσα της μπορούμε να προσθέσουμε άλλα υπάρχοντα control τα οποία θα μπορούν να μας δώσουν την ευελιξία και την λειτουργία που θέλουμε.

Τα δεδομένα που θα εμφανίζει αυτό module βρίσκονται στον πίνακα Announcements. Έτσι αφού το περιεχόμενο του είναι αποθηκευμένο μόνο σε ένα πίνακα θα είναι όλο ομοιογενές. Οπότε με ευκολία η data list μπορεί να το επαναλάβει δίνοντας σε κάθε ανακοίνωση την ίδια εμφάνιση και λειτουργία

Τα control που υπάρχουν μέσα στην data list είναι ένα image button το οποίο μας οδηγεί στην σελίδα επεξεργασίας του module, ένα label το οποίο είναι δεμένο (bind) με κάποιο πεδίο από την πηγή δεδομένων που έχει τραβήξει το περιεχόμενο από τον πίνακα και τα εμφανίζει διαδοχικά μια μια τις τιμές του πεδίου αυτού και ένα link button το οποίο πατώντας το εμφανίζονται περισσότερες πληροφορίες για την ανακοίνωση που έχουμε επιλέξει στέλνοντας μας στην σελίδα της ανακοίνωσης όπου υπάρχει ολοκληρωμένο το περιεχόμενο .

Το περιεχόμενο εμφανίζεται μέσω των καταλλήλων δεσιμάτων (bindings) που έχουν γίνει μεταξύ της πηγής δεδομένων και των control. Ο μόνος έλεγχος που θα πρέπει να κάνουμε παραπάνω είναι αν ο χρήστης έχει δικαίωμα να επεξεργαστεί το περιεχόμενο αυτού του module και να εμφανίσουμε την το κουμπί που τον στέλνει στην σελίδα επεξεργασίας.

Πατώντας το κουμπί επεξεργασίας οδηγούμαστε στην σελίδα (Announcements) edit.aspx σελίδα.

Announcement σελίδα επεξεργασίας

Εdit.aspx

Λεπτομέρειες Ανακοίνωσης

Τίτλος:	<input type="text"/>	Πρέπει να εισάγετε έναν έγκυρο τίτλο
Σύνδεσμος:	<input type="text"/>	
Σύνδεσμος (κινητού):	<input type="text"/>	
Περιγραφή:	FCKeditor V2 - DescriptionField	Πρέπει να εισάγετε μια έγκυρη περιγραφή
Λήξη:	<input type="text"/>	Πρέπει να εισάγετε μια έγκυρη ημερομηνία Πρέπει να εισάγετε μια έγκυρη ημερομηνία

[Ενημέρωση](#) [Ακύρο](#) [Διαγραφή](#)

Δημιουργήθηκε από τον [CreatedBy] στις [CreatedDate]

Αυτή είναι η σελίδα επεξεργασίας των ανακοινώσεων. Όπως μπορούμε να διακρίνουμε υπάρχουν όλα τα απαραίτητα πεδία για την καταχώρηση των πληροφοριών που μας χρειάζονται. Ξεκινώντας από την κορυφή της σελίδας μπορούμε να διακρίνουμε τα πεδία για την καταχώρηση του τίτλου, του συνδέσμου σε περίπτωση που χρειάζεται να παραπέψουμε σε κάποια άλλη τοποθεσία, του κυρίου περιεχομένου, τότε λήγει η ανακοίνωση αλλά και από ποιον δημοσιεύτηκε και τότε.

Στο πεδίο του κυρίου περιεχομένου συναντάμε και πάλι τον Fck editor με τον οποίο μπορούμε να μορφοποιήσουμε το κείμενο της ανακοίνωσης. Το πεδίο

Σύστημα διαχείρισης περιεχομένου ιστού

λήξης ορίζει από πότε και μετά η ανακοίνωση θα πέψη να ισχύει και άρα μετά από αυτήν την ημερομηνία δεν θα προβάλλεται άλλο.

Τα controls που είναι τοποθετημένα στο δεξιό μέρος της σελίδας ονομάζονται field validators είναι και αυτά control του visual studio και κάνουν τους απαραίτητους ελέγχους για σε κάποιο πεδίο που τους έχουμε ορίσει για το αν έχει σωστή τιμή, αλλιώς κατά την καταχώρηση στον πίνακα της βάσης θα προκύψει σφάλμα.

Παράδειγμα αν στο πεδίο ημερομηνία λήξης αντί για ημερομηνία βάλουμε κείμενο θα ενεργοποιηθεί ο validator και θα μας ενημερώσει ότι έχουμε βάλει λάθος τιμή.

Calendar



Σε αυτό το module δεν έχουμε κάποια ιδιαίτερη λειτουργία. Αλλά ούτε έχουμε κάποιο κομμάτι κώδικα για κάποιο συγκεκριμένο σκοπό. Αυτό το module χρησιμεύει μόνο για να μπορούμε να δουμε την τρέχουσα ημερομηνία. Το module αυτό έχει στο κέντρο του ένα Calendar control του visual studio το οποίο είναι ρυθμισμένο να παίρνει από τον server την ημερομηνία της ημέρας

Clone Module

Ένα άλλο module το οποίο πραγματικά μας λύνει τα χέρια είναι το Clone module. Με το module αυτό μπορούμε να χρησιμοποιούμε το περιεχόμενο κάποιας σελίδας σε όποια άλλη σελίδα επιθυμούμε. Στον πυρήνα του δεν περιέχει κάποιο Control και αυτό γίνεται επειδή θα χρησιμοποιήσει θα προστεθεί στο περιεχόμενο του το άλλο module που του έχουμε πει να αντιγράψει με ότι άλλο έχει αυτό μέσα του.

Στην βάση μας υπάρχει ένας πίνακας που ονομάζεται ModuleSettings. Σε αυτόν τον πίνακα είναι καταχωρημένα κατά αντιστοιχία ποιο Clone module αντιγράφει ποιο module. Κάθε φορά που αντιγράφουμε κάτι με το clone module σε κάποια σελίδα πρέπει να της προσθέσουμε εκ νέου ένα clone module και στον πίνακα ModuleSettings καταχωρείται η συσχέτιση του με το περιεχόμενο που αντιγράφει.

Η λογική που υπάρχει στον κώδικα που εκτελείται μέσα στον Clone module είναι η εξής. Όταν φορτώνεται η σελίδα και πάει να γίνει render το module ήδη ξέρουμε το ID του καθώς και τα υπόλοιπα στοιχεία του. Τότε ελέγχουμε αν υπάρχει κάποια καταχώρηση στον πίνακα ModuleSettings. Αν εντοπίσουμε εγγραφή τότε από τον πίνακα modules παίρνουμε τα απαραίτητα στοιχεία του module που αντιγράφεται καθώς και το περιεχόμενο του. Μόλις γίνουν γνωστά αυτά τα δεδομένα τότε προσθέτουμε στο clone module το module αντιγράψαμε.

Clone Module Edit Page



Αυτή είναι η σελίδα επεξεργασίας του clone module. Τα βασικά στοιχεία που περιέχει είναι δυο λίστες επιλογής ή drop down list. Αυτές οι λίστες επιλογής είναι control του visual studio στα οποία μπορούμε να κρατάμε και να προβάλλουμε κάποιες τιμές ως επιλογές.

Όταν δημιουργείται η σελίδα επεξεργασίας η μια λίστα γεμίζει από τα ψευδώνυμα των σελίδων που υπάρχουν στην εφαρμογή από τον πίνακα Tabs και η άλλη λίστα γεμίζει από τα module που περιέχει αυτή η σελίδα .

Τα drop down list μας επιτρέπουν να κρατάμε δυο τιμές σε δυο αντίστοιχα πεδία τιμών που έχει. Το ένα πεδίο ονομάζεται TextField και το άλλο value Field. Στο πρώτο κρατάμε το ψευδώνυμο της σελίδας στο άλλο το ID της που είναι καταχωρημένη στην βάση. Επίσης και στην άλλη λίστα κρατάμε το όνομα το module και του ID αντίστοιχα.

Η διαδικασία που πρέπει να γίνει εδώ έχει ως εξής. Αφού έχουμε αποφασίσει το περιεχόμενο ποιου Module θα επαναλαμβάνεται εδώ πρέπει να το εντοπίσουμε από τις λίστες και να το καταχωρήσουμε στον πίνακα ModuleSettings. Οπότε από την λίστα των σελίδων πάμε και επιλέγουμε την σελίδα που προϋπάρχει το module. Μόλις επιλέξουμε την σελίδα αυτόματα γεμίζει η λίστα των module από τα Module που υπάρχουν σε αυτήν την σελίδα και κάνουμε την επιλογή μας .

Πατώντας αποθήκευση καταχωρούμε στον πίνακα ModuleSettings το ID του clone module που επεξεργαζόμαστε με το ID του module της λίστας των Module που μόλις επιλέξαμε και τελειώνει η διαδικασία της αντιγραφής.

Εδώ ίσος μας γενάτε η απορία αν μπορούμε να αντιγράψουμε clone module από clone module, μα φυσικά και γίνεται αφού και το Clone module είναι module κάποιας σελίδας και είναι καταχωρημένο στον πίνακα modules άρα θα εμφανιστεί στη λίστα επιλογών και μέσω του πίνακα ModuleSettings μπορούμε να εντοπίσουμε ποιο περιεχόμενο αντιγράφει.

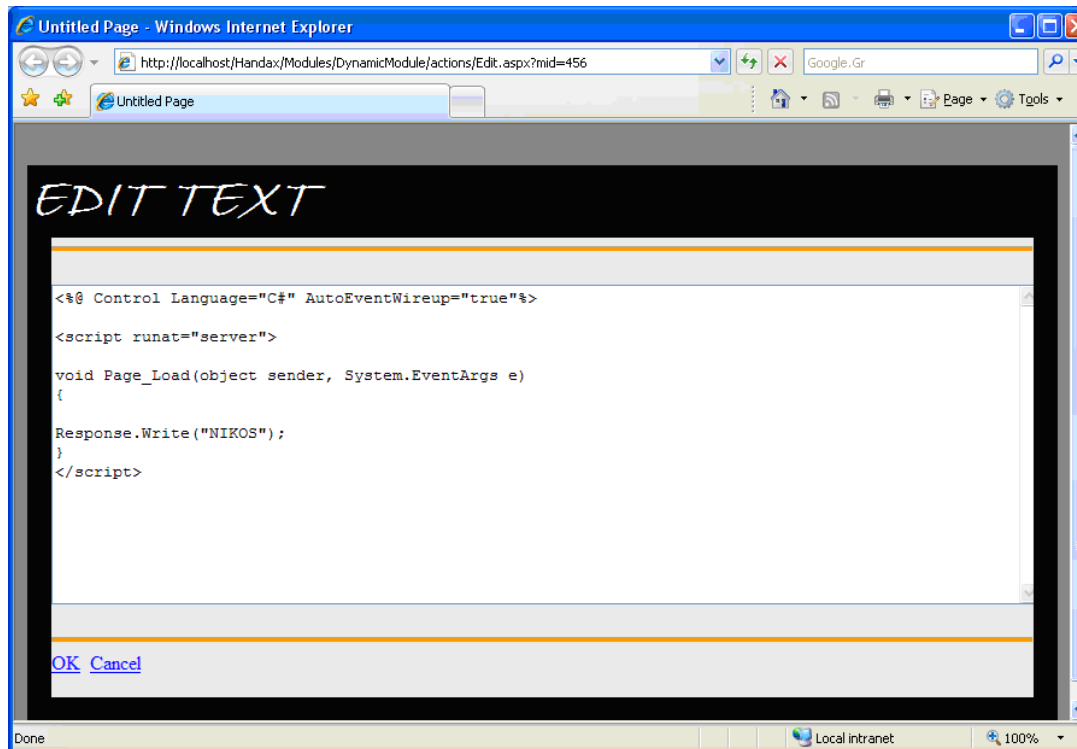
Dynamic Module

Πόσες είναι οι φορές που θέλουμε να κάνουμε κάτι στην εφαρμογή μας όπως να της προσθέσουμε παραπάνω ευελιξία, να βελτιώσουμε μια υπάρχουσα υπηρεσία ή να κάνουμε κάτι δικό μας καινούργιο όπως θα μας βόλευε καλύτερα ή να το κάνουμε ακριβώς όπως θα μας άρεσε.

Αν ρωτάτε εμένα πάρα πολλές αλλά δεν μπορούσα να επέμβω στην εφαρμογή γράφοντας ούτε μια γραμμή κώδικα για να κάνω αυτό που πραγματικά επιθυμούσα . Αυτό ακριβώς έρχεται να λύσει το Dynamic Module να μας λύσει τα χέρια και να μας δώσει την δυνατότητα να φτιάχνουμε καινούργια πράγματα από την αρχή στο σημείο που επιθυμούμε, μιας και είναι module και μπορούμε να το προσθέσουμε οπουδήποτε, όπως εμείς τα θέλουμε χωρίς να χρειάζεται να κάνουμε debug ολόκληρη την εφαρμογή από την αρχή.

Το πώς ακριβώς δημιουργείται ένα dynamic module και πως λειτουργεί θα το αναλύσουμε στην σελίδα επεξεργασίας του. Αφού ουσιαστικά εκεί γίνεται όλη η δουλειά και εδώ απλά εμφανίζονται τα αποτελέσματα της. Το dynamic module δεν περιέχει κάτι μέσα του αφού εμείς θα φτιάξουμε ότι μας χρειάζεται απλά μας παρέχει τον χώρο και τον τρόπο να γράψουμε τον κώδικα που θα υλοποιεί τις επιθυμίες μας.

Dynamic Module Edit Page



Παραπάνω μπορούμε να δούμε την σελίδα επεξεργασίας του dynamic module. Όπως διακρίνεται δεν έχουμε τίποτα παραπάνω από ένα textbox για να μπορούμε να επικολλήσουμε τον κώδικα αφού πρώτα τον έχουμε δοκιμάσει στο visual studio ή σε κάποιο άλλο περιβάλλον ανάπτυξης που να μπορούμε να κάνουμε debug τον κώδικα μας για τυχόν λάθη.

Ο κώδικας τοποθετείται στο Module αλλά με τη διαφορά ότι θα είναι σε script μορφή αφού τον εισάγουμε στο source αρχείο του module και όχι στο ενσωματωμένο αρχείο c sharp. Με αυτό τον τρόπο κάθε φορά που στέλνεται το module από το server στον client γίνεται compile από την μηχανή asp net που τρέχει στον server. Έτσι δεν χρειάζεται να αναλάβουμε εμείς το compile, αλλιώς θα έπρεπε να κάνουμε compile ολόκληρη την εφαρμογή για να ενσωματωθούν οι αλλαγές στα εκτελέσιμα dll's αρχεία της. Έτσι με μόνο μέλημα μας το σωστά συντακτικά γραμμένο κώδικα μπορούμε να κάνουμε τις τροποποιήσεις και τις αλλαγές μας όποτε εμείς κρίνουμε απαραίτητο.

Τώρα θα πρέπει πούμε πως γίνεται η διαδικασία. Την στιγμή που εισάγουμε στην σελίδα που επεξεργαζόμαστε ένα dynamic module κρατάμε το Module ID που του δίνει η βάση. Με το που έχουμε πάρει το ID στα χέρια μας μέσα σε ένα φάκελο στην εφαρμογής στο σημείο που είναι αποθηκευμένο το πρότυπο dynamic module δημιουργούμε ένα αρχείο με την κατάληξη ascx και όνομα ότι και το ID που έχουμε. Τα αρχεία ascx είναι user controls για τα οποία έχουμε ήδη μιλήσει. Όταν δημιουργηθεί το αρχείο αυτό είναι τελείως άδειο οπότε έχουμε την δυνατότητα να γράψουμε ότι θέλουμε μέσα του εδώ τον κώδικα για τις απαιτήσεις μας χωρίς να περιοριζόμαστε από κάτι. Επειδή το πώς θα ενεργήσει ο κώδικας του module εξαρτάται από την asp machine του server κατά κάποιο τρόπο εμάς

λύνονται τα χέρια μας. Το μόνο που έπρεπε να κάνουμε είναι να αποθηκεύσουμε από το text box τον κώδικα που έχουμε γράψει στο αρχείο ID.ascx. Αυτό δεν έχει καμία ιδιαίτερη δυσκολία γιατί μπορούμε να το εξεργαστούμε σαν ένα κανονικό αρχείο κειμένου.

Ίσως εδώ να έχουμε παραλείψει ότι αφού μπορούμε να γράψουμε κανονικό κώδικα σε αντικειμενοστραφή γλώσσα μπορούμε να έχουμε και visual περιβάλλον με όλα τα διαθέσιμα controls του visual studio και του .net τα οποία μπορούν αν δημιουργούν events και να αλληλεπιδρούν με τον χρήστη και όλη την υπόλοιπη εφαρμογή.

Είναι το πιο πολύτιμο module από την άποψη ότι μπορούμε χωρίς δυσκολία να προσθέτουμε και να αναπτύσσουμε νέες ενέργειες και λειτουργίες στην εφαρμογή μας χωρίς να χρειάζεται να την ανεβοκατεβάζουμε στον server.

Page Edit



Επειδή είναι κουραστικό και χρονοβόρο να ψάχνουμε ανάμεσα σε σελίδες και επιλογές για να κάνουμε μία απλή εργασία προσπαθήσαμε με αυτό το module να συγκεντρώσουμε τις πιο συνηθισμένες ενέργειες που γίνονται στην σελίδα στην κορυφή της και να είναι καθ' όλη την διάρκεια της εργασίας μας ορατές ώστε να είναι άμεσα προσπελάσιμες.

Το module αυτό τοποθετείται στην σελίδα μας μόνο αν ο χρήστης έχει δικαιώματα επεξεργασίας στην σελίδα. Το module αυτό ενσωματώνει τρεις βασικές λειτουργίες την επεξεργασία σελίδας την προσθήκη σελίδας με γονική σελίδα την τρέχουσα και την διαγραφή της σελίδας από την βάση και την εφαρμογή. Αυτό που πραγματικά κάνει το Module αυτό είναι να μας κάνει redirect στις αντίστοιχες σελίδες Tab settings όπου και γίνεται η επεξεργασία της σελίδας στην delete page και στην add page.

Για την σελίδα tab settings θα αναφερθούμε εκτενεστέρα παρακάτω μιας και έχει αρκετές λειτουργίες πάνω της. Όταν μας γίνεται redirect στην σελίδα delete page στο url συμπεριλαμβάνεται και το ID της σελίδας που επεξεργαζόμαστε. Εκεί αφού διαβάσουμε από το url το ID εκτελούμε όλα τα απαραίτητα Sql ερωτήματα προς την βάση για να σβήσουμε όλα τα δεδομένα της σελίδα συν όλα τα δεδομένα των module που την απαρτίζουν. Μόλις πλοηγηθούμε προς την add page πρέπει να ξαναδιαβάσουμε το ID. Μόλις μάθουμε το ID της σελίδας από το url με το κατάλληλο ερώτημα προς την βάση εισάγουμε μία καινούργια σελίδα στην βάση με parent Id το ID της σελίδας. Το parent id δηλώνει ότι η καινούργια σελίδα θα κρέμεται στο menu κάτω από την σελίδα που επεξεργαζόμασταν. Οι τιμές που παίρνει η σελίδα για να καταχωρηθεί στην βάση είναι συγκεκριμένες αλλά και μοναδικές για κάθε σελίδα. Αφού τελειώσει η καταχώρηση της νέας σελίδας από την σελίδα add page γίνεται redirect προς την σελίδα tab settings. Εκεί μπορεί ο χρήστης να επεξεργαστεί εκ νέου την σελίδα και να της δώσει ότι τιμές επιθυμεί αλλά και να τις αλλάξει την γονική σελίδα.

Tab Menu



Εδώ θα ασχοληθούμε με το μενού της εφαρμογής. Δηλαδή το module που συγκεντρώνει τις σελίδες της εφαρμογής ή εξωτερικές συνδέσεις από την εφαρμογή. Το module αυτό δεν είναι τίποτα άλλο από ένα user control που στον πυρήνα του έχουμε προσθέσει το control του visual studio Menu

Το μενού δεν είναι τίποτα άλλο από μια συλλογή αντικειμένων μενού που έχουν κάποιες τιμές και ονομασίες. Ο χρήστης τοποθετώντας τον κέρσορα πάνω στα διάφορα αντικείμενα του μενού και κάνοντας κλικ επάνω τους μπορεί να προσπελάσει τους πόρους της εφαρμογής.

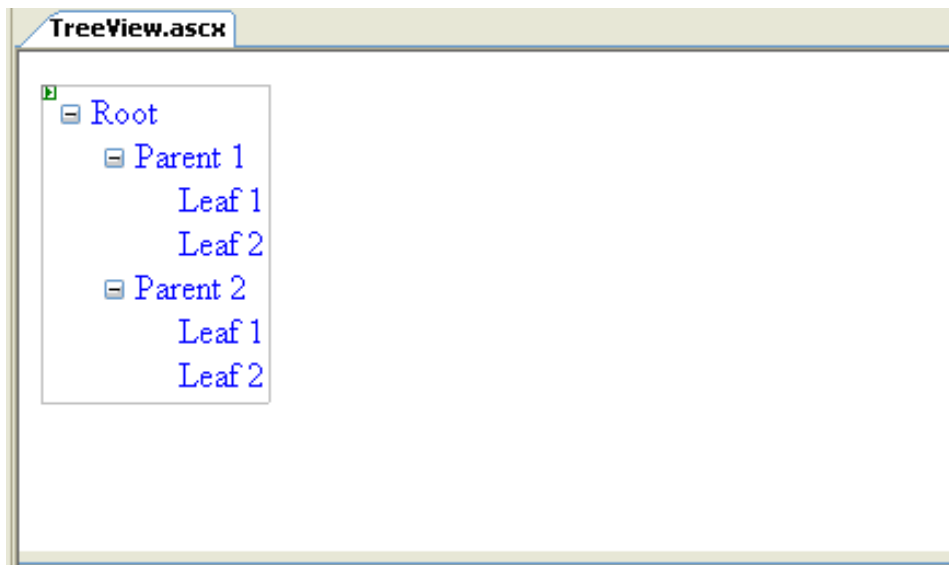
Το ζήτημα εδώ ήταν πως θα καταφέραμε να γεμίσουμε το μενού με τα αντικείμενα του και να τους δώσουμε τι τιμές τους. Ο ένας τρόπος είναι χειροκίνητα, να είχαμε δηλαδή κατά την δημιουργία της εφαρμογής με το χέρι καταχωρήσει στο μενού τα αντικείμενα του αλλά αυτό δεν θα μας επέτρεπε να καταχωρήσουμε καινούργιες σελίδες. Ο άλλος τρόπος είναι να έχουμε ένα αρχείο xml που ανάλογα τι αλλαγές γίνονται στην εφαρμογή να το ενημερώνουμε και να το ορίζουμε ως πηγή δεδομένων στο μενού και να γεμίζει αυτόματα. Ωμως και αυτή η επιλογή δεν μας δίνει την ευελιξία που θέλουμε.

Έτσι καταλήξαμε ένα γεμίζουμε τα αντικείμενα του μενού με δυναμικό τρόπο. Λέγοντας δυναμικό τρόπο εννοούμε να τραβάμε όλα τα δεδομένα από τον πίνακα Tabs και κάθε σελίδα που είναι αποθηκευμένη σε αυτόν τον πίνακα να μετατρέπεται με προγραμματιστικό τρόπο σε αντικείμενο μενού.

Η διαδικασία εδώ λοιπόν είναι η εξής: με το κατάλληλο sql ερώτημα γεμίζουμε ένα πίνακα με τα δεδομένα του πίνακα Tabs. Έπιτα με μια ρουτίνας επανάληψης τρέχουμε τα στοιχεία του πίνακα. Πριν τα τοποθετήσουμε στο μενού κοιτάμε αν έχουν κάποιες γονικές σελίδες για να τους τις προσθέσουμε ως αντικείμενα μενού από κάτω τους. Γι αυτό το λόγο χρειαζόμαστε και άλλη μια ρουτίνα για να το ελέγξουμε αυτό. Αν ένα κάποια σελίδα έχει στο πεδίο Parent Id το ID κάποιας άλλης σελίδας τότε είναι παιδί προς αυτή. Όσες σελίδες δεν έχουν Parent Id δηλαδή είναι ίσο με το μηδέν πάει να πει ότι δεν είναι παιδί σε κάποια άλλη σελίδα και τοποθετείτε στην κορυφή του μενού ώστε όσες σελίδες εξαρτούνται από αυτή να μπουν από κάτω της.

Έτσι με κατορθώνουμε να ενημερώνουμε αυτόματα το μενού κάθε φορά που φορτώνεται και με κάθε αλλαγή να είναι εμφανή από την πρώτη στιγμή.

TreeView



Ένα άλλο module πλοήγησης που χρησιμοποιούμε είναι το tree view. Το module αυτό βασίζεται στο control του visual studio tree view. Με αυτό το module μπορούμε να εμφανίσουμε με δένδρική μορφή τα δεδομένα και να περιηγηθεί ο χρήστης ανάμεσα τους.

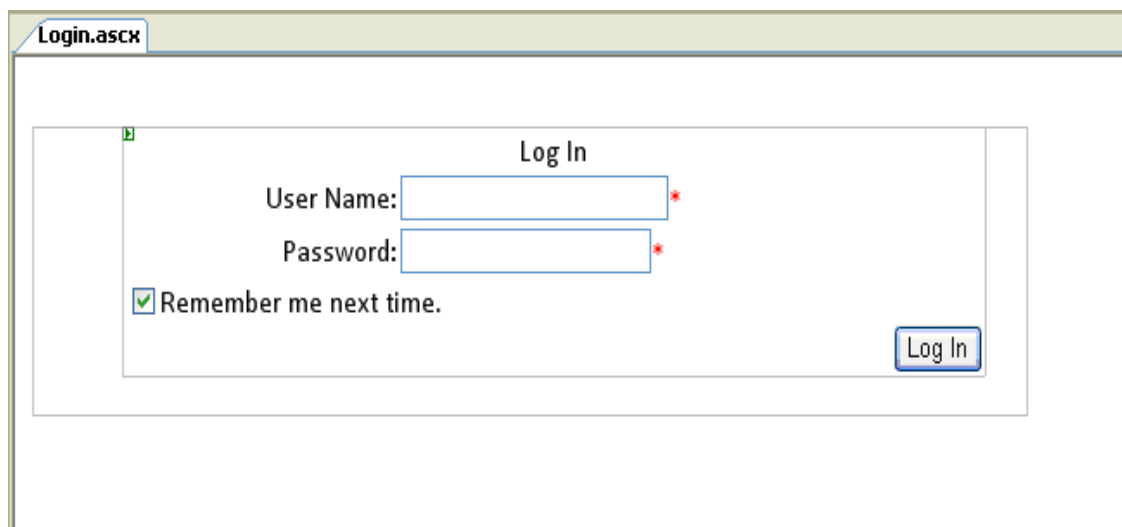
Όπως και το μενού υπάρχει η εξάρτηση γονέα παιδιού για το πώς θα εμφανιστούν στην τελική τους μορφή. Το βασικό σημείο αυτού του control είναι οι κομβοί. Ως κύριοι κόμβοι μπαίνουν οι σελίδες που δεν ανήκουν σε κάποια άλλη ως γονικές και κάτω από αυτές εξαρτώμενες από τον κύριο κόμβο όλες όσες είναι παιδιά της.

Το γέμισμα και αυτού του module γίνεται από μία αναδρομική συνάρτηση μόνο τώρα που τα δεδομένα δεν τα τραβάμε από τον πίνακα Tabs αλλά από τον πίνακα Trees.

Το tree view συνήθως μας βοηθάει στην πλοήγηση σε ενότητες θεμάτων που υπάρχουν μέσα στην εφαρμογή και να μεταπηδάμε από το ένα θέμα στο άλλο με μεγαλύτερη ευκολία παρά να ψάχνουμε μέσα στο κεντρικό και μεγαλύτερο μενού της εφαρμογής που είναι ευκολότερο να χαθούμε.

Επίσης μπορεί να φανεί χρήσιμο στο να το έχουμε σε όλες τις σελίδες της εφαρμογής μας στο αριστερό μέρος η όπου αλλού εμείς επιθυμούμε και να έχουμε άμεση σύνδεση με τα πιο σημαντικά σημεία της εφαρμογής.

Login



The screenshot shows a web browser window with a tab titled 'Login.ascx'. Inside the browser, there is a form titled 'Log In'. The form contains two text input fields: 'User Name:' and 'Password:'. Both fields have a red asterisk to their right, indicating they are required. Below the password field is a checkbox labeled 'Remember me next time.' which is checked. A 'Log In' button is located at the bottom right of the form.

Όπως έχουμε πει στην εφαρμογή μας υπάρχουν κατηγορίες χρηστών με διαφορετικά δικαιώματα, διαφορετικές αρμοδιότητες στο περιεχόμενο και με δικαιώματα επεξεργασίας του περιεχόμενου που προβάλλεται. Έτσι για να έχουν πρόσβαση οι χρήστες στο περιεχόμενο που έχουν αυξημένες αρμοδιότητες πέρα από το περιεχόμενο που είναι κοινό για όλους θα πρέπει με κάποιο τρόπο να τον ταυτοποιήσουμε και να τους εξουσιοδοτήσουμε να έχουν πρόσβαση στο περιεχόμενο που είναι για συγκεκριμένους χρήστες αλλά και να τους αποδώσουμε τις λειτουργίες που έχουν στο περιεχόμενο αυτό.

Η ταυτοποίηση γίνεται με κάποιους κωδικούς που έχει ο κάθε χρήστης και μέσω αυτού του module ελέγχουμε την εγκυρότητα των κωδικών αυτών. Έτσι με αυτό το module γίνεται ο έλεγχος αν υπάρχει ο χρήστης στην βάση με τον αντίστοιχο κωδικό.

Όλη η διαδικασία της ταυτοποίησης του χρήστη γίνεται στη κλάση UsersDB την οποία θα αναλύσουμε παρακάτω σε άλλη ενότητα. Εδώ μπορούμε να πούμε με ποιο τρόπο ξέρουμε το ποιος χρήστης είναι συνδεδεμένος στην εφαρμογή. Μόλις γίνει η ταυτοποίηση από την βιβλιοθήκη του .NET τοποθετούμε στα cookies του χρήστη το authentication cookie της εφαρμογής μας στο οποίο είναι τοποθετημένες όλες οι πληροφορίες που μας ενδιαφέρουν. Οπότε κάθε φορά που ο χρήστης ζητεί κάποιο περιεχόμενο της εφαρμογής ή κάποια διαχειριστική λειτουργία συγκρίνουμε τα δικαιώματα του χρήστη με τα δικαιώματα του περιεχομένου ή λειτουργίας για να την προσπελάσει. Αν έχει ο χρήστης τα απαραίτητα δικαιώματα τότε προωθείται σε αυτό που ζήτησε αλλιώς τον στέλνουμε στην σελίδα άρνηση πρόσβασης περιεχομένου.

Το control που περιέχει το module είναι το login control του visual studio. Το επιλέξαμε αυτοί το control και δεν το σχεδιάσαμε μόνοι μας προς χάρην ευχρηστίας αλλά και γιατί μπορεί να παρέχει ασφάλεια σε sql injection και έλεγχο στις τιμές των πεδίων αλλά και διαχειριστεί το authentication cookie

Σύστημα διαχείρισης περιεχομένου ιστού

Εδώ τελειώνει η παρουσίαση των Modules και η ανάλυση τους. Παρακάτω θα προσπαθήσουμε να παρουσιάσουμε τον φάκελο admin που συμπεριλαμβάνει όλα τα διαχειριστικά της εφαρμογής, τον φάκελο components που περιέχει όλες τις βοηθητικές κλάσεις της εφαρμογής αλλά και κλάσεις από τις οποίες κληρονομούν ιδιότητες διάφορα αντικείμενα που χρησιμοποιούμε στην εφαρμογή μας.

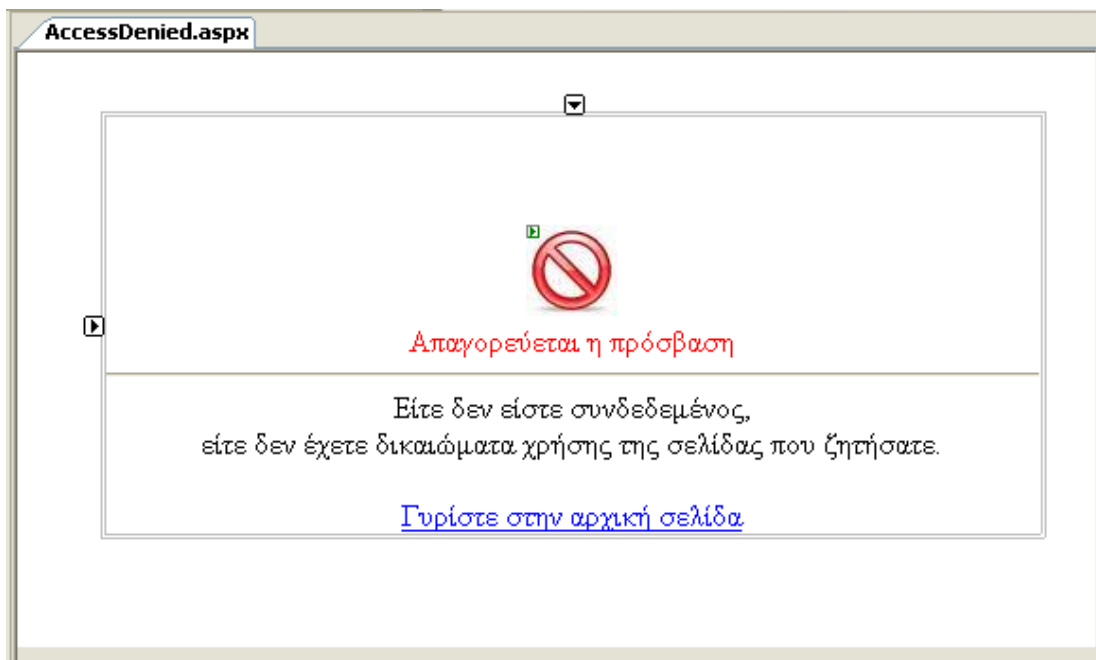
Φάκελος admin

Αυτός ο φάκελος είναι η καρδιά της εφαρμογής μας. Όλα όσα χρειαζόμαστε να για να διαχειριστούμε την εφαρμογή και το περιεχόμενό της είναι συγκεντρωμένα μέσα στον φάκελο αυτό. Επεξεργασία σελίδων και module δικαιώματα πρόσβαση στην σελίδα και module γιατί κάποιος χρήστης μπορεί να έχει πρόσβαση στην σελίδα αλλά να μην επιτρέπεται να δει κάποιο από τα modules της σελίδας.

Επίσης εδώ βρίσκονται η σελίδα επεξεργασίας για τους καταχωρημένους χρήστες των ρόλων και των δικαιωμάτων που έχει ο κάθε ρόλος.

Ας τα πάρουμε όμως από την αρχή να τα παρουσιάσουμε ένα ένα

Access Denied



Αυτό το μήνυμα θα το βλέπουμε όποτε προσπαθούμε να προσπελάσουμε κάποιο πόρο της εφαρμογής που δεν έχουμε τα απαραίτητα δικαιώματα. Μας ειδοποιεί και μας ενημερώνει ότι δεν είμαστε εξουσιοδοτημένοι να δούμε το περιεχόμενο που ζητήσαμε και μας προτρέπει με ένα σύνδεσμο να γυρίσουμε στην σελίδα όπου ήμασταν.

Αυτός ο σύνδεσμος είναι δεμένος (bind) με το Url της σελίδας όπου ήρθαμε εδώ μέσω της εντολής `UrlReferrer` που απομνημονεύει την προηγούμενη κατάσταση μας.

AddNewTab

Αυτή η σελίδα δεν έχει κάποιο γραφικό στο περιεχόμενο της γιατί χρησιμοποιείται ως σύνδεσμος από το module επεξεργασίας της σελίδας και τη σελίδα Tab settings.

Δουλεία αυτής της σελίδας είναι να δημιουργεί μια νέα σελίδα και να την καταχωρεί στην βάση και μετά να προωθεί το Id που μόλις έδωσε η βάση στην νέα σελίδα στη σελίδα tab settings για την τελική επεξεργασία και για την τελική της μορφή.

Σαν δεδομένα μπορεί να πάρει από το Url το id της σελίδας που θα είναι η καινούργια σελίδα παιδί. Μόλις βρει αυτό το id καταχωρεί στην βάση την καινούργια σελίδα με προκαθορισμένες τιμές από μας και μετά μας κάνει redirect στην σελίδα tab settings για περαιτέρω επεξεργασία.

Delete Module

Ούτε και αυτή η σελίδα έχει γραφικό περιβάλλον γιατί δεν κάνει κάποια παραπάνω ενέργεια από το να σβήνει ένα module το περιεχόμενο του και ότι άλλο σχετίζεται με αυτό από την βάση. Για να αρχίσουμε την διαδικασία διαγραφής του module θα πρέπει να γνωρίζουμε το id μιας και αυτό το καθορίζει μονοσήμαντα στην βάση. Το ID ακόμα μια φορά το παίρνουμε από το url.

Ένα βήμα όμως πριν την διαγραφή πρέπει να βρούμε όλους τους πίνακες της βάσης για να το αναζητήσουμε σε αυτούς. Θα μπορούσαμε να τους έχουμε στατικά αποθηκευμένους σε ένα data table στην εφαρμογή αλλά το συγκεκριμένο data table δεν θα ήταν δυνατό μετά να ενημερωθεί σε περίπτωση που προσθέσουμε ένα νέο πίνακα. Με το που έχουμε συγκεντρώσει όλους τους πίνακες της βάσης αρχίζουμε να εκτελούμε ερωτήματα διαγραφής απέναντι τους με όρισμα το ID του module.

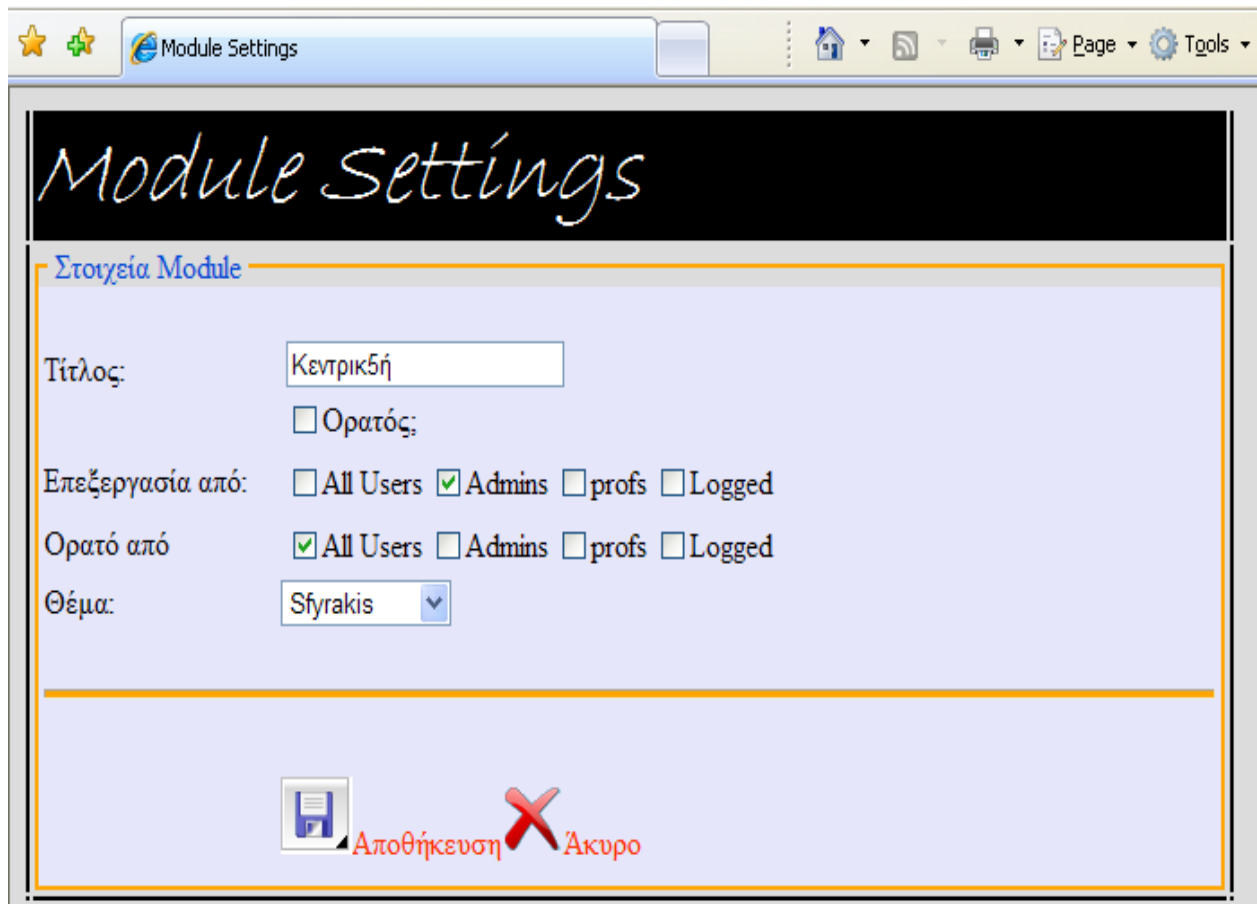
Την διαγραφή την κάνουμε με αυτόν το τρόπο γιατί μερικοί πίνακες δεν συνδέονται με τους υπόλοιπους και έτσι οι αλυσιδωτές (cascade) διαγραφές δεν μπορούν να ενεργήσουν και να διαγράψουν όλες τις συσχετίσεις του module αυτού στους πίνακες.

Logoff

Η σελίδα αυτή όπως φανερώνει το όνομα της αποσυνδέει τους συνδεδεμένους χρήστες με την εφαρμογή. Ούτε και αυτή η σελίδα δεν έχει γραφικό περιβάλλον γιατί δεν χρειάζεται να πλοηγηθούμε σε αυτή την σελίδα να πατήσουμε κάποιο κουμπί για να αποσυνδεθούμε. Αλλά πολύ απλά μπορούμε από ένα link να πλοηγηθούμε σε αυτή την σελίδα και μόλις αρχίσει να εκτελείται ο κώδικας στο page load να γίνουν όλες οι απαραίτητες ενέργειες.

Για να αποσυνδεθεί κάποιος χρήστης από την εφαρμογή θα πρέπει να εκτελέσουμε την εντολή sign out από την βιβλιοθήκη του .net για να διαγραφούν όλα τα αποθηκευμένα στοιχεία κατά την ταυτοποίηση του από το σύστημα αλλά και να σβήσουμε όσες μεταβλητές τύπου session έχουμε κρατήσει στον server για την δίκη μας διευκόλυνση στις ενέργειες που εκτελούμε στην εφαρμογή μας και πρέπει να της έχουμε συγκεντρωμένες από σελίδα σε σελίδα.

Module Settings



Module Settings

Module Settings

Στοιχεία Module

Τίτλος:

Ορατός:

Επεξεργασία από: All Users Admins profs Logged

Ορατό από All Users Admins profs Logged

Θέμα:

Αποθήκευση Άκυρο

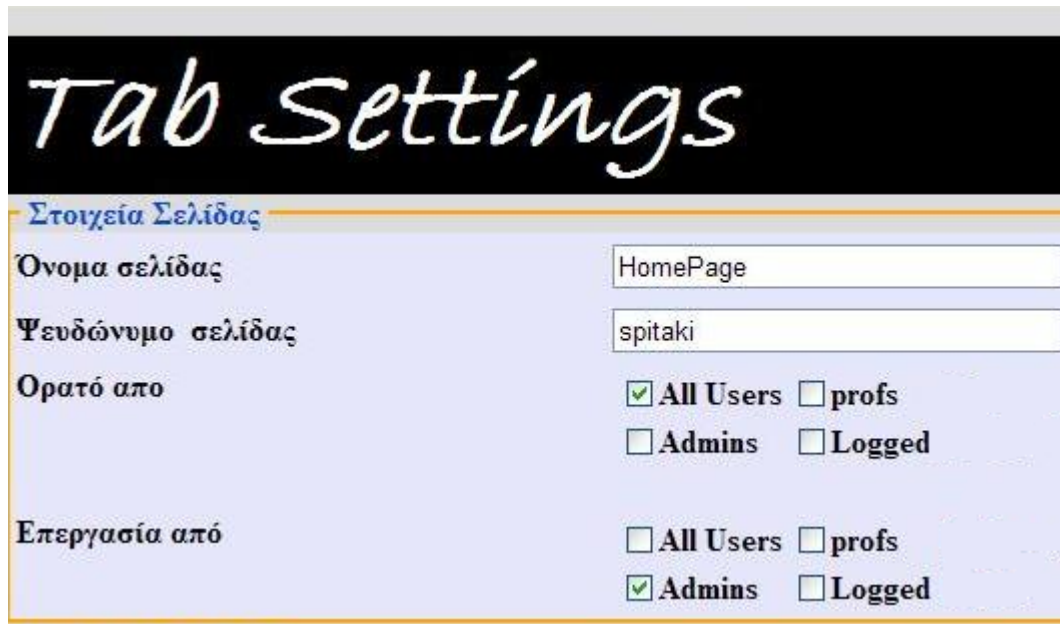
Εδώ μπορούμε να δούμε την σελίδα που διαχειριζόμαστε τις ιδιότητες των διαφόρων module που συμπεριλαμβάνονται στην εφαρμογή μας.

Η σελίδα αυτή αποτελείται από πέντε controls του visual studio με τα οποία δίνουμε ρυθμίζουμε τις τιμές και τις ιδιότητες του κάθε module.

Από το πρώτο control της σελίδας το πλαίσιο κειμένου μπορούμε να δώσουμε την ονομασία του module. Ακριβώς από κάτω υπάρχει ένα check box το οποίο ρυθμίζει αν φαίνεται ο τίτλος του module ή όχι όταν προστεθεί στην σελίδα. Αμέσως από κάτω υπάρχουν οι ρόλοι που μπορούν να έχουν οι χρήστες στην εφαρμογή μας. Η κατηγορίες που συμπεριλαμβάνουν τους ρόλους είναι δυο η μία κατηγορία ορίζει το ποιος θα μπορεί να βλέπει το module και η άλλη κατηγορία ποιος θα το επεξεργάζεται. Οι ρόλοι των χρηστών εμφανίζονται στα δύο check box lists οι οποίες γεμίζουν από τον πίνακα roles της βάσης. Όταν γίνεται το module render στην σελίδα τότε συγκρίνουμε τους ρόλους του Module με τον ρόλο του τρέχοντος χρήστη και ανάλογα προσθέτουμε τα αντίστοιχα actions στο module ή δεν το προσθέτουμε καν στην σελίδα αν δεν έχει ο χρήστης δικαίωμα για να δει το περιεχόμενο του. Τέλος η τελευταία ρύθμιση που γίνεται στο module είναι αν θα του προσθέσουμε κάποιο θέμα. Το θέμα όπως έχουμε πει είναι ένα user control το οποίο κληρονομεί και αυτό από την κλάση module control και περικλείει το module ένα γραφικό περίβλημα.

TabSettings

Σε αυτήν την σελίδα επεξεργαζόμαστε οτιδήποτε έχει σχέση με την σελίδα. Από το όνομα της το ψευδώνυμο της τους πολλούς που θα ενεργούν πάνω της και πως μέχρι το ποια module θα συμπεριλαμβάνει σε ποια θέση και με ποια σειρά. Ας τα δούμε όμως ένα ένα.



Στοιχεία Σελίδας

Όνομα σελίδας:

Ψευδώνυμο σελίδας:

Ορατό απο: All Users profs
 Admins Logged

Επεργασία από: All Users profs
 Admins Logged

Αυτή είναι η ενότητα που διαχειριζόμαστε τα βασικά στοιχεία της σελίδα όνομα ψευδώνυμο και ποιο ρόλοι θα έχουν δικαίωμα να την επεξεργαστούν και να την προσπελάσουν. Ο τρόπος είναι ακριβώς ο ίδιος με την σελίδα των module settings. Το ψευδώνυμο της σελίδας μας χρησιμεύει στο url rewrite γιατί εκεί βάζουμε ένα πιο φιλικό όνομα που είναι πιο εύκολο για τον χρήστη να το θυμάται και να φτιάξει το url όπως θέλει για να πλοηγηθεί με ευκολία όπως έχουμε πει στην ενότητα url rewrite.



Ιδιότητες Module

Module:

Τίτλος: Ορατός;

Θέση στη σελίδα:

Επεξεργασία από: All Users Admins profs Logged

Ορατό από: All Users Admins profs Logged

Θέμα:

Προσθήκη Module 

Σε αυτό το κομμάτι της σελίδας Tab settings μπορούμε να δούμε το πώς προσθέτουμε ένα καινούργιο module στη σελίδα μας. Είναι εύκολο να διακρίνουμε στην αρχή αυτής της ενότητας την λίστα επιλογής που μας εμφανίζει όλα τα module που είναι διαθέσιμα από την εφαρμογή μας.

Τα module είναι τοποθετημένα σε ένα συγκεκριμένο μέρος στην εφαρμογής σε φακέλους με το όνομα τους. Έτσι με τις εντολές αρχείων του .net μπορούμε να διαβάσουμε ένα ένα τους φακέλους από αυτό το μέρος και να ξέρουμε πόσα και ποια modules έχουμε, και με αυτόν τρόπο γεμίζουμε την λίστα επιλογής. Έτσι όταν προσθέσουμε ένα καινούργιο module στην εφαρμογή μας ο διαχειριστής θα ενημερώνεται αυτόματα για την ύπαρξη του και θα μπορεί να το χρησιμοποιήσει αμέσως. Αυτό μπορεί να γίνει γιατί όπως έχουμε πει ο κώδικας των module είναι ανεξάρτητος της εφαρμογής και δεν ενεργούν αυτόνομα αλλά μέσα στην εφαρμογή πάντα.

Μετά από τις ρυθμίσεις ρόλων επιλέγουμε ποιο χρήστες θα μπορούν να τα βλέπουν ποιο όχι ποιοι θα μπορούν να τα επεξεργάζονται και ποιοι όχι πράγματα για τα οποία έχουμε μιλήσει παραπάνω.

Αφού πρώτα έχουμε επιλέξει σε ποιο σημείο της σελίδας θα τοποθετηθεί το module το προσθέτουμε στην σελίδα πατώντας το κουμπί προσθήκη module.με το που πατήσουμε το κουμπί το module και ότι στοιχεία του έχουμε ρυθμίσει αποθηκεύονται στην βάση και εμφανίζεται στο ακριβώς επόμενο τμήμα της σελίδας όπου γίνεται η στοίχιση των modules αλλά και η ταξινόμηση τους.



Σε αυτό το τμήμα της σελίδας επεξεργασίας μπορούμε να αλλάξουμε το module από το ένα μέρος της σελίδας στο άλλο αλλά και να ρυθμίσουμε το πώς θα εμφανίζονται με ποια σειρά στο μέρος της σελίδας που έχουν τοποθετηθεί. Επίσης μας δίνεται η δυνατότητα να διαγράψουμε άμεσα το Module αν δεν επιθυμούμε την ύπαρξη του αλλά και να παρωθηθούμε κατ ευθείαν στην σελίδα επεξεργασίας module settings και να κάνουμε τις αλλαγές μας στις ρυθμίσεις του.

Οι λίστες που υπάρχουν στην σελίδα γεμίζουν από τον πίνακα modules που υπάρχει στην βάση με κριτήριο τιμή είναι καταχωρημένη στο πεδίο Pane name. Όταν μετακινηθεί ένα module από το ένα σημείο της σελίδας στο άλλο αλλά αλλάζουμε την τιμή του Pane name.

Επίσης το module στην βάση έχει και ένα άλλο πεδίο το Module order. Σύμφωνα με αυτό το πεδίο εμφανίζονται στην σελίδα. Το module με το μικρότερο Tab order εμφανίζεται πρώτο και το module με το μεγαλύτερο Tab order

Σύστημα διαχείρισης περιεχομένου ιστού

εμφανίζεται τελευταίο. Ανάλογα με το που θέλουμε να τοποθετήσουμε το Module πιο πάνω ή πιο κάτω στην σειρά εμφάνισης ανταλλάσσουμε την τιμή του πεδίου Tab order του module που μετακινείται με την τιμή του πεδίου Tab order του module που βρίσκεται στην θέση που θέλουμε να το μετακινηθεί.

Και προχωράμε στο τελευταίο κομμάτι της σελίδας Tab settings.



Αυτό το τμήμα της σελίδας μας επιτρέπει να επιλέξουμε σε ποιο σημείο θα εμφανίζεται η σελίδα στο κεντρικό μενού της εφαρμογής μας. Μπορούμε να επιλέξουμε να εμφανίζεται ως κύρια σελίδα στο μενού στην κορυφή του ή να είναι παιδί σε κάποια σελίδα.

Αυτό γίνεται από την επιλογή γονική σελίδα. Αν δεν επιλέξουμε κάτι σε αυτή την λίστα επιλογής τότε η σελίδα είναι κύρια και τοποθετείται στην αρχή του μενού αλλιώς αν επιλέξουμε κάποια σελίδα τότε θα τοποθετηθεί κάτω από την σελίδα της επιλογής μας αυτή η λίστα μας εμφανίζει όλες της σελίδες που υπάρχουν στην εφαρμογή με τα ψευδώνυμα τους, ο μόνος περιορισμός που έχουμε είναι ότι δεν μπορούμε φυσικά να επιλέξουμε τον εαυτό της σελίδας για γονική της σελίδα

Η δεύτερη επιλογή που μας δίνεται εδώ είναι να επιλέξουμε κάποια σελίδα ως κύρια. Με αυτό εννοούμε ότι τα modules της κύριας σελίδας θα φορτωθούν πρώτα στην σελίδα και μετά θα ακολουθήσουν τα modules της σελίδας μας. Αυτό μας δίνει ευελιξία όταν χρειάζεται να έχουμε μια πλειάδα σελίδων με την ίδια μορφή ή το μεγαλύτερο ποσοστό του περιεχομένου τους να είναι ίδιο. Έτσι για να αποφύγουμε την επανάληψη του ίδιου περιεχομένου φτιάχνουμε μία σελίδα πρότυπο ως προς τις άλλες και απλά την προσαρμόζουμε στις σελίδες που επιθυμούμε.

Admin Page

Αυτή η σελίδα είναι η καρδιά της εφαρμογής όλες οι βασικές λειτουργίες βρίσκονται εκεί όπως και οι χρήστες με τους ρόλους.

Η σελίδα την συγκροτούν τα admin modules που μέσα από αυτά γίνεται η διαχείριση της εφαρμογής. Η σελίδα περιλαμβάνει τα εξής Modules.

Manage Users ,Security Roles, Tabs, Site Settings.

Ας τα δούμε όμως ένα ένα τα modules αυτά ξεχωριστά

Tabs

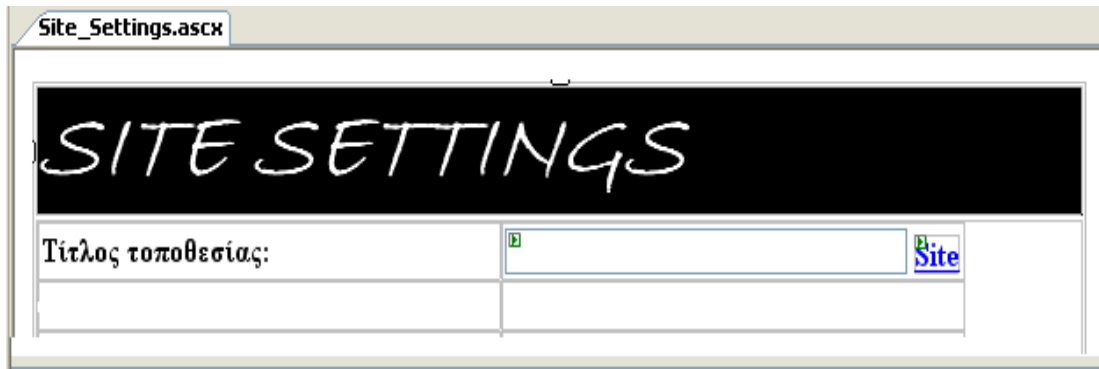


Εδώ έχουμε όλα τις σελίδες της εφαρμογής μαζεμένες σε tree view αλλά και όλες τις ενέργειες που μπορούν να δεχτούν οι σελίδες μας.

Για να αλληλεπιδράσουμε με κάποια σελίδα θα πρέπει πρώτα να την εντοπίσουμε στο tree view και να την επιλέξουμε. Μετά μπορούμε να την επεξεργαστούμε πατώντας στο εικονίδιο επεξεργασίας και οδηγούμαστε στην σελίδα Tabs Settings για την οποία έχουμε μιλήσει παραπάνω. Μπορούμε ακόμα να την μετακινήσουμε ανάμεσα στις άλλες σελίδες . Αν μετακινήσουμε κάποια σελίδα που βρίσκεται σε κύριο κόμβο μαζί της θα μετακινηθούν και τα παιδιά της. Η μετακίνηση της σελίδας γίνεται πάλι όπως και στα module ανταλλάσσοντας την τιμή του πεδίου Tab order.

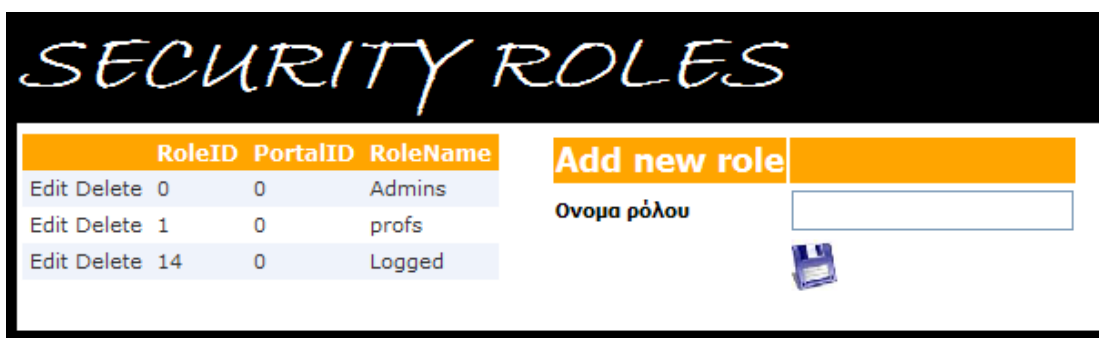
Μι ακόμα λειτουργία που ενσωματώνεται εδώ είναι η προσθήκη νέας σελίδας. Η λογική είναι ακριβώς η ίδια με το τρόπο που έχουμε αναλύσει παραπάνω. Αν έχουμε επιλέξει κάποια σελίδα στο tree view τότε η καινούργια σελίδα θα προστεθεί ως γονική αλλιώς ως κύρια στο μενού μας.

Site Settings



Με αυτό το module επεξεργαζόμαστε τον τίτλο της εφαρμογής δηλαδή το πως θα ονομάζεται και επίσης από τον σύνδεσμο site πλοηγούμαστε στην επεξεργασία του Banner εμφανίζεται στην σελίδα admin.

Security Roles



Το module αυτό μας δίνει την δυνατότητα αν προσθέσουμε κάποιο νέο ρόλο στην εφαρμογή μας, ο ρόλος αυτός αποθηκεύεται στον πίνακα Roles.

Επίσης από εδώ μπορούμε να επεξεργαστούμε το όνομα κάποιου ρόλου που ήδη υπάρχει ή ακόμα και να τον σβήσουμε.

Σε περίπτωση που αλλάζουμε το όνομα κάποιου ρόλου δεν προκύπτει κάποιο πρόβλημα αφού ενημερώνεται ο πίνακας Users-Roles με αποτέλεσμα όσοι χρήστες ανήκαν στον παλιό ρόλο τώρα να ανήκουν στον καινούργιο. Εκτός αυτό και οι όσες επιλογές είχαμε σε διάφορα module για επιλογή ρόλων ενημερώνονται αυτόματα αφού γεμίζουν δυναμικά με τις επιλογές τους

Εάν σβήσουμε το όλο όσοι χρήστες είχαν πρόσβαση σε κάποιο module ή σελίδα δεν την έχουν πια αφού και από το module θα έχει αφαιρεθεί αυτή τιμή από το πεδίο πρόσβασης

Manage Users

Manage Users

Users	Properties	
----- ▾	User ID	[lblUserId]
	User	<input type="text"/> Δεν έχετε δώσει όνομα χρήστη
	Role	Unbound ▾ Επιλέξτε ρόλο
	Password	<input type="password"/>
	Confirm Password	<input type="password"/> Οι κωδικοί πρόσβασης δεν ταιριάζουν
	Email	<input type="text"/>
<input type="button" value="↺"/> <input type="button" value="💾"/> <input type="button" value="✖"/>		

Από αυτό το module μπορούμε να επεξεργαστούμε τους καταχωρημένους χρήστες να προσθέσουμε κάποιο νέο χρήστη ή και να διαγράψουμε κάποιον υπάρχοντα χρήστη.

Για να γίνει η επεξεργασία των δεδομένων κάποιου χρήστη που είναι καταχωρημένος στην βάση πρέπει πρώτα να τον επιλέξουμε από την λίστα επιλογών που βρίσκεται στο αριστερό μέρος εκεί εμφανίζονται όλοι οι υπάρχοντες χρήστες και ανάλογα επιλέγουμε όποιον επιθυμούμε. Μόλις επιλέξουμε τον χρήστη που επιθυμούμε στο αριστερό μέρος του Module εμφανίζονται τα καταχωρημένα στοιχεία του χρήστη. Το μόνο στοιχείο που δεν εμφανίζεται είναι ο κωδικός που χρησιμοποιεί κατά την είσοδο του στην εφαρμογή για ευνόητους λόγους. Αμέσως μετά μπορεί να αλλάξει ότι δεδομένο επιθυμεί

Για να καταχώρηση κάποιο νέο χρήστη αρκεί να εισάγει το όνομα του στο πεδίο του ονόματος μόλις γίνει αυτό θα ενεργοποιηθούν και οι υπόλοιπες επιλογές και θα μπορεί να γίνει η καταχώρηση.

Αν κάποιος τώρα προσπαθήσει να αλλάξει τα στοιχεία κάποιου χρήστη ου ήδη υπάρχει στη βάση δεν θα το καταφέρει γιατί οι αλλαγές εξαρτώνται από το ID που δίνουμε και όχι από το όνομα και μόνο από αυτό.

Εδώ τελειώσαμε και με τον φάκελο admin που περιέχει τα κυριότερα διαχειριστικά κομμάτια της εφαρμογής. Παρακάτω θα αναλύσουμε τα περιεχόμενα του φακέλου

Σύστημα διαχείρισης περιεχομένου ιστού

Components που περιέχει όλες τις κλάσεις που εξαρτώνται τα διάφορα αντικείμενα της εφαρμογής αλλά και για τις βοηθητικές κλάσεις που μας βοηθάνε να μην επαναλαμβάνουμε κώδικα για τον έλεγχο κάποιων καταστάσεων αλλά και για την εκτέλεση κάποιων λειτουργιών.

Φάκελος Components

Κλάσεις

UsersDB

Εδώ είναι το μέρος στο οποίο γίνονται όλοι οι έλεγχοι για τα δικαιώματα του χρήστη όπως αν υπάρχει στην βάση καταχωρημένος αν έχει πρόσβαση σε κάποιο πόρο και ότι άλλο σχετίζεται αυξημένα δικαιώματα χρήστη.

Η κλάση αυτή έχει πέντε μέθοδος που πολλές φορές η μια έχει ανάγκη τα αποτελέσματα της άλλης για να εκτελέσει τον κώδικα της

Η πρώτη μέθοδο που συναντάμε είναι η Login(), αυτή η μέθοδος καλείται από το module login για να ελέγξει τα στοιχεία που έδωσε ο χρήστης. Έχοντας ως δεδομένα το username και password που έχει δώσει ο χρήστης εκτελούμε το κατάλληλο ερώτημα στην βάση για να πάρουμε το όνομα του μιας και το username που χρησιμοποιούμε στην εφαρμογή μας είναι το email του χρήστη. Εάν το ερώτημα επιστρέψει κάποιο αποτέλεσμα σημαίνει ότι ο χρήστης υπάρχει και προχωρήσει η διαδικασία στο login module για την είσοδο του χρηστή στην εφαρμογή, αλλιώς εμφανίζεται μήνυμα ότι δεν υπάρχει ο συγκεκριμένος χρήστης.

Η δεύτερη μέθοδος είναι η IsInRole(string roles) και ελέγχει αν ο χρήστης ανήκει σε κάποιο συγκεκριμένο ρόλο. Αυτή η μέθοδος μας χρησιμεύει για να συγκρίνουμε το ρόλο του χρήστη με τους ρόλους κάποιου module που πάει να προστεθεί στην σελίδα για να ελέγξουμε αν μπορεί να το δει ο χρήστης. Επιπλέον με αυτή την μέθοδο μπορούμε να ελέγξουμε αν ο χρήστης έχει δικαίωμα να προσπελάσει την σελίδα που ζήτησε.

Η μέθοδος που μόλις αναλύσαμε συνεργάζεται με την μέθοδο GetUserRole() για να πάρει το ρόλο που αναζητά για να κάνει την σύγκριση και επιστέφει αληθές ή ψευδές αποτέλεσμα.

Η τελευταία μέθοδος είναι η User Role η οποία επιστρέφει το όνομα του χρήστη στην μέθοδο Login()

Announcements

Με αυτή την κλάση γίνονται όλοι οι απαραίτητοι έλεγχοι από και προς το module Announcements για να αποθηκεύσει ή να προβάλει το περιεχόμενο που είναι καθορισμένο να προβάλει. Όλες οι διαγραφές ενημερώσεις και καταχωρήσεις των νέων ανακοινώσεων γίνονται από αυτή την κλάση.

Επίσης μέσα από της μέθοδος της κλάσης Announcements γεμίζουν και τα controls στο module Announcements που είναι δεμένα με κάποιο τύπο αποτελέσματος.

GlobalAsaxHelper

Σε αυτή την κλάση γίνεται ο απαραίτητος έλεγχος για την εγκυρότητα του μονοπατιού για να προσπελάσουμε κάποιο πόρο μέσω του Url rewrite. Περιέχει μια μέθοδο που η οποία επιστρέφει αληθές ή ψευδές αποτέλεσμα.

DataCache

Μέσω αυτής κλάσης αποθηκεύουμε στην cache του server τα δεδομένα από την βάση που χρησιμοποιούμε σε αρκετά σημεία στην εφαρμογή. Με αυτήν την κλάση μπορούμε να κρατήσουμε στη μνήμη ολόκληρους πίνακες από την βάση δεδομένων και με τις κατάλληλες μεθόδους που περιλαμβάνονται μέσα στην μέθοδο να πάρουμε το πλήθος και τον συνδυασμό των αποτελεσμάτων που επιθυμούμε.

Όλη αυτή την διαδικασία την κάνουμε γιατί είναι αρκετά χρονοβόρο να ρωτάμε κάθε φορά την βάση για κάποια τιμή και επίσης φορτώνεται παραπάνω το σύστημα ενώ έτσι τα δεδομένα είναι προσβάσιμα άμεσα και σε λιγότερο χρόνο.

Ας φανταστούμε μία εφαρμογή με αρκετές σελίδες πόσο χρόνο θα ήθελε για να δημιουργήσει το μενού κάθε φορά αν περίμενε τα δεδομένα από την βάση

HtmlTextDB

Αυτή η κλάση αναλαμβάνει όλες τις απαραίτητες ενέργειες για την ανάκτηση του html περιεχομένου για λογαριασμό του Html module όσο και για την αποθήκευση του στον πίνακα HTMLTEXT στην βάση. Το περιεχόμενο που αποθηκεύεται είναι η τιμή που έχει ο κειμενογράφος Fck editor την δεδομένη στιγμή.

ModuleAction

Μέσω αυτής της κλάσης προσθέτουμε σε κάθε module της ενέργειες που του αναλογούν. Οι ενέργειες αυτές έχουμε αναφέρει ότι είναι επεξεργασία, εκτύπωση, διαγραφή. Αυτή η κλάση διαβάζει το xml αρχείο που περιέχει τις πληροφορίες για τις ενέργειες του κάθε module και όποια βρει την προσθέτει στο ειδικό μέρος στο θέμα του περιβάλλει το module.

TabHandler

Στην μέθοδο αυτή έχουμε τις μεθόδους που μας χρειάζονται για να επεξεργαστούμε και για να φορτώνουμε τις σελίδες της εφαρμογής μας. Οι μέθοδοι που περιέχει ελέγχουν αν η σελίδα έχει πρότυπη σελίδα από ποιους ρόλους είναι προσπελάσιμη αλλά και επεξεργάσιμη.

Theme

Αυτή η κλάση κληρονομεί από την κλάση του συστήματος UserControl οπότε όλα τα θέματα που έχουμε είναι και αυτά UserControls αλλά επιπλέον με τις μεθόδους της κλάσης Theme.

Σε συνεργασία με την κλάση ModuleAction η κλάση Theme προσθέτει τα action στα modules. Απλά η κλάση ModuleAction έχει πιο πολύ ως εργασία να εντοπίσει τα actions κάθε Module αλλά η Theme έχει την ευθηνή να προσθέσει τα action στα θέματα του κάθε Module μιας και αυτά κληρονομούν από αυτήν.

Τα θέματα για μας δεν είναι τίποτα άλλο από UserControls τα οποία δεν εκτελούν κανένα κομμάτι κώδικα μέσα τους απλά έχουν χρώματα και εικόνες τα οποία περιβάλλουν τα modules που προστίθενται σε αυτά για αισθητικούς λόγους

Επίλογος

Σε αυτή την εργασία παρουσιάσαμε πως λειτουργεί ένα σύστημα διαχείρισης περιεχομένου και πως μπορούν διάφορες ανεξάρτητες μεταξύ τους ενότητες να συνδυαστούν για να παρουσιαστεί το περιεχόμενο το οποίο προορίζεται η εφαρμογή μας.

Διαπιστώσαμε επίσης την ευελιξία και την ευκολία στην διαχείριση που μπορεί να μας δώσει η βάση δεδομένων που βρίσκεται πίσω από το κυρίως πρόγραμμα

Επιπλέον είδαμε ότι το περιεχόμενο μπορεί να κατηγοριοποιηθεί και να προσπελάζεται από επιλεγμένους χρήστες

Οι δυνατότητες όμως ενός δυναμικού συστήματος δεν σταματάνε εδώ. Είναι πλήρως επεκτάσιμο χάρη στο δυναμικό module που συμπεριλαμβάνεται στην εφαρμογή. Επίσης μπορούν εύκολα να προστεθούν νέα module όποτε κρίνεται αυτό απαραίτητο ή το απαιτούν οι ανάγκες μιας καινούργιας εφαρμογής .

Έτσι η το σύστημα και η εφαρμογή μπορούν να είναι πάντα ενημερωμένα και να ακολουθούν τις εξελίξεις της τεχνολογίας χωρίς κανένα πρόβλημα