COMAS(Content Management System)

A THESIS SUBMITTED TO

THE INSTITUTE OF SCIENCE

OF

THE UNIVERSITY OF HALİÇ

BY

MURAT ÇATAL

ISTANBUL, FEBRUARY 2006

© Bu projenin bütün hakları
Haliç Üniversitesi Bilgisayar Mühendisliği Bölümü'ne aittir.

PREFACE

Computers are one of the essentials of our life. Data is being evaluated more swift, more reliable and productive. In every area in order to get speed to make decisions, processing data flow and documentation we come against with computers. These days another developing field of computer system is Internet. It is very easy to get information and data from wherever you are in the world by Internet. It is very usual to use the Internet.

After arising of Internet, we came across another concepts "Content Management System (CMS)". CMS is an important tool for the site owners. Because, After the initial implementation by CMS Admin, the site owners can easily manage, update and maintain the site. CMS is built to administer and publish mid to large sized web sites, corporate and vertical portals.

In computing, a content management system (CMS) is a system used to organize and facilitate collaborative creation of documents and other content. A CMS is frequently a web application used for managing websites and web content, though in many cases, content management systems require special client software for editing and constructing articles.

In this project, I aim to enable users to manage, update, maintain, publish their sites over Internet.

I want to thank to manager of my project Mr. Prof. Dr. Ali Okatan for guiding and her precious suggestions during the development steps of the project.

ABSTRACT

In this thesis, I aim to enable users to manage, update, maintain, publish their sites over Internet. .

In this project, I planned to supply oppurtunities for the owners of the Internet site such as easy and quick implementation , modular and elastic structure, powerful technology, easily management of websites and web content, adding new modules, user-friendly interfaces, region&boxes system.

This is a software application that makes it possible for non-technical users to publish content to a website.

In the development duration of the project, I determined the alternatives and chosen one of them by caring of efficiency, compatibility and usefulness. According to this, I decided to use ASP.NET as web programming style, VB.NET and JavaScript as scripting languages, Microsoft Office Access as database management system, Windows 2000 server operating system and Internet Information Server in the ASP.NET development duration.

ÖZET

Tezde, son kulllanıcıya kendi sitelerini yönetebilme, güncelleyebilme, bakım yapabilme ve yayınlayabilme imkanı verecek içerik yönetim sistemi oluşturmayı amaçlandı.

Projede, kolay ve çabuk uygulama, moduler ve elastik yapı, güçlü teknoloji, web sitelerini ve web içeriğini kolay yönetme, yeni modüller ekleme, kullanıcı dostu ortamlar ve bölge&kutu sistemi gibi bir çok etken ile, sistem sahiplerine önemli imkanların sunulması hedeflenmiştir.

İçerik yönetim sistemi, teknik olmayan kullanıcılara site içeriği yönetme olanağı sağlayan bir yazılım uygulamasıdır.

Proje yapımında, gerekli fizibilite çalışmaları yapılarak, kullanılacak teknolojiler belirlenmiştir. Uygulama geliştirme platformu olarak ASP.NET, programlama dili olarak VB.NET, veritabanı olarak Microsoft Office Access, işletim sistemi olarak Windows 2000 Server ve sunucu olarak ISS kullanılmasına karar verilmiştir. XML Web Service, XML, XSLT, HTML, DHTML ve daha bir çok teknoloji proje yapımında kullanılmıştır.

1. INTRODUCTION

With this project it is aimed to administer and publish mid to large sized web sites, corporate and vertical portals online. This project is called as COMAS(Content Management System)

COMAS is built to administer and publish mid to large sized web sites, corporate and vertical portals. After the initial implementation by COMAS Admin, the site owners can easily manage, update and maintain the site.

COMAS has been developed with VB.Net and its powerful architecture. N-tier architecture has been used with presentation, process and database layers.

Features :

- Easy and quick implementation
- Modular and elastic structure allows updates and developments
- With its parametric structure, it can be customized to fulfill various needs
- Powerful technology. COMAS has been developed with Microsoft.NET technology.
- Allows complete control over websites and intranets
- Eliminates dependency on third parties to manage your site
- New modules can be easily integrated
- No software other than a web browser is needed.
- Every internet user can easily use COMAS, no specific knowledge is required
- System can be easily managed
- Permission wizards allow many administrators with specific permissions
- Oracle, SQL Server , MS Access, MySQL databases are supported
- Region and Content Box system allows full control over every page.

Basic Components

COMAS Content Management System includes the following basic modules:

- **Homepage Management:** Most of the websites have a special cover page and then a template for interior pages. With this assumption, the home page management and interior page management has been separated and homepage has its unique template.
- **Interior Pages Management:** This component is used to take the full control over all the interior pages of the site. Picture Management: Picture management allows users to resize the pictures, rotate them and also write text on them. With this component, the small edits to the pictures are done on the fly without using any custom image processing software.
- **User Management:** A web site can be management by more than one webmaster. To allow multi admin feature, a user management section with permission management is added.
- **Permission Management:** This module is used to set the permissions of the administrators. All the tasks in COMAS can be restricted or granted access via this component.
- **Box Management:** COMAS content management system is based on regions and boxes which will be explained in detail in the next sections. Box management is used to create boxes and modify predefined boxes.

Modules

COMAS's modular structure allows development of custom components according to various needs. With the addition of new modules, the web site or the portal contains all the

features and functions you need. For example, a Poll module can be added to COMAS and from the administration panel you can add, delete or view the results. Below are some modules already developed :

- Announcement and news module
- Poll Module
- FAQ Module
- E-Support Module
- Membership Module
- Article Management Module

Daily Menu for Meals Module

COMAS continuously can be updated for new modules and custom modules are easily and quickly can be developed specific to any needs.

Region and Box System

COMAS, different from many content management systems, uses two concepts Regions and Boxes to allow the management of content with as much flexibility as possible.

Homepage and Interiors pages' templates are divided into regions link the column in a newspaper. However, the regions do not have to be vertical, horizontal regions can also be created.

Boxes are the content units added to the regions. Each box has its own design and content can be entered into it very easily. Unlimited number of boxes can be added to a region and unlimited types of boxes can be created. The usage of boxes ensures the consistency of the look of content within the site.

Regions are formed while setting up the initial site. The following example shows 4 regions;

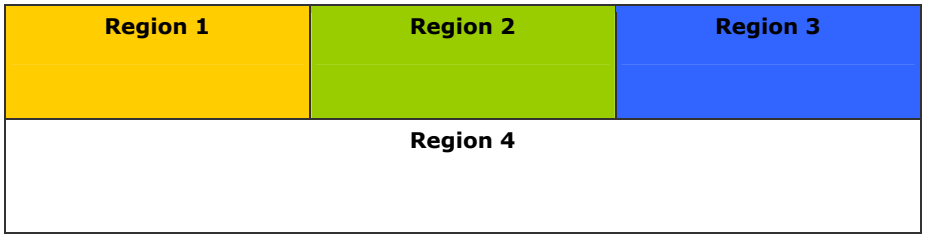


Figure 1 A template with 4 regions

Boxes can be created later but COMAS Admin creates the initial boxes when implementing the system. Boxes can have different types like a paragraph, a table, a table with a picture and so on. COMAS Admin also creates boxes for the modules it creates for you to add it to any page you want (ex: For the user module, a login box is created to be used anywhere within the site)

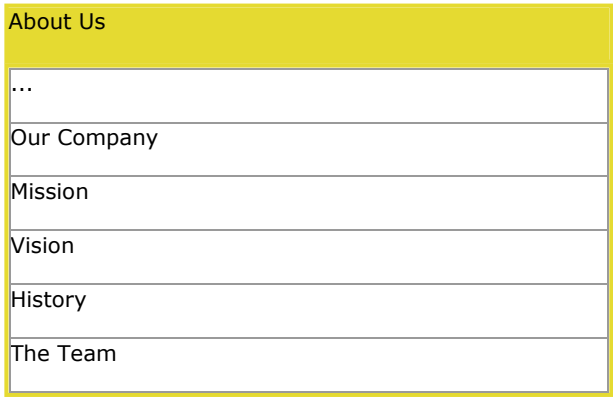


Figure 2 Box Sample

As will be explained in details in the following pages it is agreed to use ASP.NET for web programming language, VB.NET and JavaScript for scripting language and Microsoft Office Access as being the database management system.

ASP.NET pages are interactive pages that run in the web browser and which create active web pages and send those pages to the online clients' browsers by means of the embedded scripts they have.

ASP.NET is a programming framework built on the common language runtime that can be used on a server to build powerful Web applications. ASP.NET offers several important advantages over previous Web development models:

ASP.NET is compiled common language runtime code running on the server. Unlike its interpreted predecessors, ASP.NET can take advantage of early binding, just-in-time compilation, native optimization, and caching services right out of the box. This amounts to dramatically better performance before you ever write a line of code.

The ASP.NET framework is complemented by a rich toolbox and designer in the Visual Studio integrated development environment. WYSIWYG editing, drag-and-drop server controls, and automatic deployment are just a few of the features this powerful tool provides.

Because ASP.NET is based on the common language runtime, the power and flexibility of that entire platform is available to Web application developers. The .NET Framework class library, Messaging, and Data Access solutions are all seamlessly accessible from the Web. ASP.NET is also language-independent, so you can choose the language that best applies to your application or partition your application across many languages. Further, common language runtime interoperability guarantees that your existing investment in COM-based development is preserved when migrating to ASP.NET.

ASP.NET makes it easy to perform common tasks, from simple form submission and client authentication to deployment and site configuration. For example, the ASP.NET page framework allows you to build user interfaces that cleanly separate application logic from presentation code and to handle events in a simple, Visual Basic - like forms processing model. Additionally, the common language runtime simplifies development, with managed code services such as automatic reference counting and garbage collection.

ASP.NET employs a text-based, hierarchical configuration system, which simplifies applying settings to your server environment and Web applications. Because configuration information is stored as plain text, new settings may be applied without the aid of local administration tools. This "zero local administration" philosophy extends to deploying ASP.NET Framework applications as well. An ASP.NET Framework application is deployed to a server simply by copying the necessary files to the server. No server restart is required, even to deploy or replace running compiled code.

ASP.NET has been designed with scalability in mind, with features specifically tailored to improve performance in clustered and multiprocessor environments. Further, processes are closely monitored and managed by the ASP.NET runtime, so that if one misbehaves (leaks, deadlocks), a new process can be created in its place, which helps keep your application constantly available to handle requests.

ASP.NET delivers a well-factored architecture that allows developers to "plug-in" their code at the appropriate level. In fact, it is possible to extend or replace any subcomponent of the ASP.NET runtime with your own custom-written component. Implementing custom authentication or state services has never been easier.

With built in Windows authentication and per-application configuration, you can be assured that your applications are secure.

2. INITIAL RESEARCH

Firstly, I examined several web Pages, content management systems for project. After I finished my investigation, I determined the most used modules.

This modules :

- Announcement and news module
- Poll Module
- FAQ Module
- E-Support Module
- Membership Module
- Article Management Module
- Picture Management Module
- Food List Management Module

At this stage the aim is to get knowledge about the system to be studied on and to define the concept of the project.

2.1. The Description Of the Content Management System

As this project is a content management system application, I believed that it will be useful to describe the content management system concepts.

CMS

In computing, a content management system (CMS) is a system used to organize and facilitate collaborative creation of documents and other content. A CMS is frequently a web application used for managing websites and web content, though in many cases, content management systems require special client software for editing and constructing articles.

This is a software application that makes it possible for non-technical users to publish content to a website.

A CMS serves as a store for a wide range of information assets, including text, image databases and so on. Information loaded into the system can be structured and output in a number of different ways. In some instances, for example, a piece of content may appear in two different places on a site.

A CMS can be reasonably simple or very complex. There is a wide range of CMS products available, developed by providers such as Microsoft, the open source community and small development companies. Choosing the right CMS can be an important decision for your organization.

Functionality

CMSs allow end-users (typically authors of some sort) to provide new content in the form of articles. The articles are typically entered as plain text, perhaps with markup to indicate where other resources (such as pictures) should be placed. The system then uses rules to style the article, separating the display from the content, which has a number of advantages when trying to get many articles to conform to a consistent "look and feel". The system then adds the articles to a larger collection for publishing.

The systems also often include some sort of concept of the workflow for the target users, which defines how the new content is to be routed around the system.

A good example would be a system for managing a newspaper. In such a system the reporters write articles in the system, which stores them in a database. Along with the article the system stores attributes, including keywords, the date and time of filing, the reporter's name, etc. The system then uses these attributes to find out—given its workflow rules—who should proofread the article, approve it for publication, edit it, etc. Later the editors can

choose which articles to include (or ignore) in an edition of the newspaper, which is then laid out and printed automatically.

More recently the term has been associated almost entirely with programs for editing web sites. Such systems span a wide variety of needs, from small systems with almost no workflow for small user-groups, to large database-based systems for running large, very active web sites such as those for a newspaper or international corporation.

A simple content management system that has seen wide use is the Wiki (or WikiWiki). Wikipedia is based on this concept, where software includes logic that allows authors (often including anonymous users) to edit the content of the site online. When the user has completed making changes, the wiki software tracks what changes you have made and updates the page. Wikis also allow more than one person to edit the target page at the same time, solving simultaneous write conflicts as they arise.

Features

Typical features include:

- Online authoring / change approval
- WYSIWYG editing
- Workflow including a review and publishing process
- Session and user management
- Indexing and Searching
- Object Storage
- Templating
- Image and URL management

Types of CMS

There are several types of content management systems:

Web CMS (WCMS) assists an organization in automating various aspects of web content creation, content management and delivery. Delivery to the web is its primary format, but many WCM systems also deliver to wireless devices.

Transactional management system or Transactional CMS (TCMS) assists an organization to manage e-commerce transactions.

Integrated CMS (ICMS) assists an organization in managing enterprise documents and content.

Publication CMS (PCMS) assists an organization in managing the publications (manuals, books, help) content life cycle.

Learning CMS (LCMS) or managed learning environment (MLE) assists an organization in managing the web-based learning content life cycle.

Enterprise CMS (ECMS) vary in their functionality. Some support both the web and publications content life cycle, while others support the web content life cycle and either transactional content or customer relationship management content.

Content management systems can also be classified by where and how the software is installed:

A traditional CMS is usually provided on a compact disc or is downloadable from the web and must be installed and configured on a server. Most CMSs nowadays are available in this format. It is beneficial because it can 'plug in' to an existing system, but may be a hinderance because it requires a great amount of system administration.

An ASP CMS is provided via an application service provider. This means you do not need to install it on a server. Most webmails are provisioned via ASP and the first CMS available via ASP was Yahoo! GeoCities. This example is fitting because historically most

tools which use ASP are either ad-based or aimed at children. However companies have been provisioning enterprise Content Management Systems to the professional and enterprise markets for some years now.

A deployed CMS is simply an ASP solution which is installed within an intranet or a user's own enviroment.

In CMS projects, the most used modules are listed like below:

- Weather Online
- Blog
- Site Log
- Search Engine Optimization
- User Defined Table
- Skin Objects
- Scoreboards
- Data Viewing, Charting, Reporting Tools
- CSS Tools
- Multi Page Content
- Gallery
- Documents
- HTML Tools
- Text Box
- Utility
- Search
- Announcements, Articles, News
- Forum

- Menu/Navigation
- Comics
- Advertising/Vendors
- FAQs
- Scheduler
- Links
- Multi-language/Localizations
- Registration
- Content Management
- RSS/Remote Content
- Events/Calendar
- Photo Album
- Games
- Review
- User Accounts
- Container Generator
- Webcam
- Quotes
- Contacts
- WAP
- Form Creation/Feedback
- Help Desk Management
- Maps
- Editor Provider

- WIKI
- Media Players
- Chat
- Guestbook
- Snippets
- Clock/Time
- Upload/Download
- Security
- Site Optimization
- Subscription Tools
- User Customization
- Tasks
- IFrame
- PDF Creation
- Podcast Creation/Syndicating
- PayPal
- Accounting/Project Management

Content Management, also known as CM, is a set of processes and technologies supporting the evolutionary life cycle of digital information. This digital information is often referred to as content or, to be precise, digital content.

This digital content life cycle consists of six primary states: create, update, publish, translate, archive and retire. For example, an instance of digital content is created by one or more authors. Over time that content may be edited. One or more individuals may provide

some editorial oversight thereby approving the content for publication. Once published that content may be superseded by another form of content and thus retired or removed from use.

Content management is an inherently collaborative process. The process consists of the following basic roles and responsibilities:

- Content Author - responsible for creating and editing content.
- Editor - responsible for tuning the content message and the style of delivery.
- Publisher - responsible for releasing the content for consumption.
- Administrator - responsible for managing the release of the content ultimately placing it into a repository so that it can be found and consumed.

A critical aspect of content management is the ability to manage versions of content as it evolves. Authors and editors often need to restore older versions of edited products due to a process failure or an undesirable series of edits.

A content management system is a set of automated processes that support the following features:

- Identification of all key users and their roles
- The ability to assign roles and responsibilities to different instances of content categories or types.
- Definition of workflow tasks often coupled with messaging so that content managers are alerted to changes in content.
- The ability to track and manage multiple versions of a single instance of content.
- The ability to publish the content to a repository in order to support the consumption of the content.

Content management systems take the following forms:

- a web content management system is software for web site management - which is often what is implicitly meant by this term
- the work of a newspaper editorial staff organization
- a workflow for article publication
- a document management system
- a single source content management system - where content is stored in chunks, within a relational database.

CMS Timeline

- 1970's - Mainframe CM and ElectronicPublishing Repository
- Atex Separation, Repository
- 1977 - PersonalComputer, TextInterface
- 1980's
- 1982 – GraphicalInterface , Xerox Parc Star, Apple Lisa WYSIWYG
- 1984 Apple Macintosh, MacWrite, MacPublisher
- 1985 PageMaker, Interleaf
- 1985 FileNet introduces Visual Workflo Workflow
- 1986 Quark Xpress
- 1990's
- 1992 Lotus Notes
- 1993 Mosaic Graphical Browser
- 1994 SoftQuad HoTMetaL Pro
- 1995 Vermeer Technologies FrontPage
- 1995 Apache WebServer

- 1995 Interleaf Cyberleaf - Internet Publishing SingleSourcePublishing
- 1995 CNET PRISM (a patented web content management system and page generation system) Personalization
- Macromedia Dreamweaver
- 1996, July. Vignette acquires CNET PRISM, integrates into StoryBuilder? and StoryServer? Web Content Management System (NewMedia Hyper Award, January 1997)
- 1996, September. SoftQuad announces HotMetaL Intranet Publisher (IBM RS/6000)
- 1996, October. Documentum announces RightSite - Industrial-Strength Web Content Management
- 1996, November. FutureTense Texture Web Publishing System (required Java-compatible browser)
- 1996 eBT Dynabase - "XML-based web content management and publishing platform" (USWeb)
- 1996, December. Inso Electronic Publishing Systems (acquires DynaBase, DynaText, DynaWeb)
- 1997 Macromedia Dreamweaver, Adobe GoLive
- 1997 Interwoven TeamSite VersionControl
- 1998 Future Tense Content Server
- 1998 TYPO3, later an open-source CMS
- 1999
- 2000 UDDI introduced by Microsoft and IBM.
- 2001 Broadvision buys Interleaf Bladerunner.
- 2001 Open Market buys Future Tense Content Server.

- 2002 FileNet acquires eGrail WCM system.
- 2002, October TikiWiki - open-source CMS
- 2002 divine acquires Open Market and Content Server.
- 2002 ASP.NET
- 2003 Red Hat acquires Ars Digita ACS.
- 2003 FatWire acquires divine Content Server.
- 2004 ASP.NET
- 2005 ASP.NET

2.2. Information of Organization

I worked with MHZ Software Company was established in Istanbul, Fatih in 2000 for selling computer parts and software applications, setting networks. MHZ Software Company want to CMS project to develop and manage their projects.

2.3. The Structure of the Existing System

The computers are used in the company has these features ;

Pentium III 450 MHz CPU

256 MB ram

20 GB hard disc

56 K fax-modem

Microsoft Windows 2000 pro operating system

Microsoft Office 2000 office tool

3. FEASIBILITY

At this stage the aim is to reveal the alternative solutions and to choose among them the most appropriate one throughout the implementation progress of the project. Feasibility can be examined in three stages; technical, economical and functional feasibility. In technical feasibility I described the requirements of the system. Which kind of hardware should I use. Economical feasibility is defined to specify the economical cost of the system. Functional feasibility is defined to specify the applicational specification of the system.

3.1. Technical Feasibility

In technical feasibility the aim is to identify the software and hardware alternatives, which can be used throughout the project implementation progress.

3.1.1. Determining the Software Tools To Be Used In the System

At this stage, the alternative software tools those can be used throughout the project implementation progress has been investigated, compared, and among them the most appropriate alternatives are chosen.

3.1.1.1. Database Management System Alternatives and Election One Among Them

3.1.1.1.1. MSSQL Server

This software runs Windows-based. Especially with the last releases it has been usable for all types of web-based applications. It is considerably successful in source usage. It can easily supply certain amounts of database needs. Microsoft implemented this system to run integrated with ASP.NET . Its interface is considerably useful and comparing to other softwares it is easier to find and train perfect staff about the application development subject.

3.1.1.1.2. MS Access

This software runs Windows-based. Comparing to SQL its usage is easier. It is useful for smaller database applications. There is no need to have any other server software because it runes with file access system. It is cheaper and learning it takes considerably shorter to learn in comparison with SQL. It is successful in source usage. But weaker in security comparing to SQL. It is really easy to find staff due to the shortness of the training duration.

3.1.1.1.3. ORACLE

Oracle database systems are platform independent. It may be stated that these systems are safer and stable in comparison with Windows based database applications. Oracle, can easily sustain the complicated structured integrated systems. The rate of finding perfect staff comparing to windows based applications is lower.

3.1.1.1.4. MYSQL

This database software is only used in Unix and Linux systems. It is optimized for the aim of Web application development. It has a perfect integration with PHP and Perl. Having a great stable structure, MYSQL serves a very easy usage for big scale database applications. Its interface is not so useful. Running with Microsoft ODBC there can be met with some problems.

Consequently regarding these criteria it is decided to implement the system's windows based database by using MS Access. But Project can support Oracle, SQL Server and MySql database by changing some features.

3.1.1.2. Scripting Language Alternatives and Election One Among Them

Unlike classic ASP, which supports only interpreted VBScript and JavaScript, ASP.NET now supports more than 25 .NET languages (including built-in support for VB.NET, C#, and JScript.NET -- no tool required), giving you unprecedented flexibility in your choice of language.

C# is a simple, modern, object oriented, and type-safe programming language derived from C and C++. It will immediately be familiar to C and C++ programmers. C# aims to combine the high productivity of Visual Basic and the raw power of C++.

C# is provided as part of Microsoft Visual Studio 7.0. In addition to C#, Visual Studio supports Visual Basic, Visual C++, and the scripting languages VBScript and JScript. All of these languages provide access to the Microsoft .NET platform, which includes a common execution engine and a rich class library. The Microsoft .NET platform defines a "Common Language Specification" (CLS), a sort of lingua franca that ensures seamless interoperability between CLS-compliant languages and class libraries. For C# developers, this means that even though C# is a new language, it has complete access to the same rich class libraries that are used by seasoned tools such as Visual Basic and Visual C++. C# itself does not include a class library.

Basically C# is supported by Microsoft.NET platforms. If C# is wanted to be used one of these must be used too:

- Server Script : If C# will be used as server scripts in ASP.NET pages, server must be Microsoft Internet Information Server (IIS) 4.0 or another third part equivalent.

- Client Script : ASP.NET pages work in all browsers -- including Netscape, Opera, AOL, and Internet Explorer. But I will use Internet Explorer.

All browsers that allow scripting support JavaScript. So, if it is wanted to write wide scaled applications for a common web site, JavaScript will be a natural election for client scripting. Due to the fact that JavaScript (and JavaScript) are both widespread, many sources that can be used as help documents can easily be found at both the bookstores and in Web.

Microsoft IIS allows using Jscript in scripting for ASP.NET pages. But the other servers might not allow this. This fact must be taken into consideration in choosing script language for Server.

Internet Explorer accepts both "JavaScript" and "JavaScript" for LANGUAGE attribute used for identifying script language in HTML code. But many other browsers can only recognize "JavaScript". So to improve adaptivity "JavaScript" is decided to be used.

Whichever you use languages(C#, VB.NET, C++, etc.) , It is not important for performance. Because, Microsoft JIT Compiler convert all kinds of code to MSIL code. So, I prefer using VB.NET for its code editor.

Regarding this factors it is admitted to use the both languages where they are more advantageous, throughout the project development progress. Therefore, for the scripts that will run script-sided VB.NET, and for the scripts that will run client-sided JavaScript is planned to be used.

3.1.1.3. Dinamic Web Tools Alternatives and ASP.NET

Why ASP.NET?

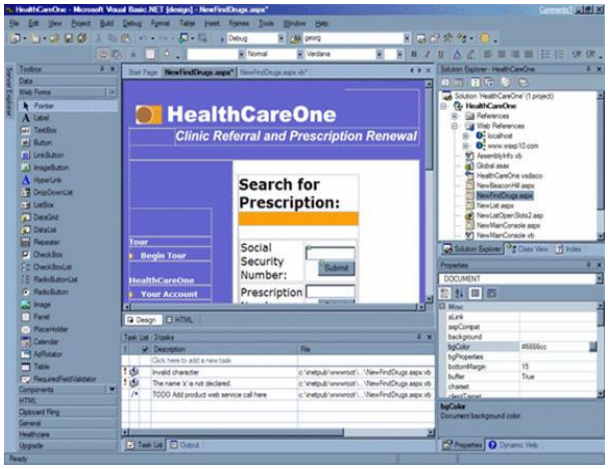
Writing dynamic, high-performance Web applications has never been easier.ASP.NET combines unprecedented developer productivity with performance, reliability, and deployment.

Developer Productivity

ASP.NET helps you deliver real world Web applications in record time.

- **Easy Programming Model:** ASP.NET makes building real world Web applications dramatically easier. ASP.NET server controls enable an HTML-like style of declarative programming that let you build great pages with far less code than with classic ASP. Displaying data, validating user input, and uploading files are all amazingly easy. Best of all, ASP.NET pages work in all browsers -- including Netscape, Opera, AOL, and Internet Explorer.

- **Flexible Language Options :** ASP.NET lets you leverage your current programming language skills. Unlike classic ASP, which supports only interpreted VBScript and JScript, ASP.NET now supports more than 25 .NET languages (including built-in support for VB.NET, C#, and JScript.NET -- no tool required), giving you unprecendented flexibility in your choice of language.



- **Great Tool Support:** You can harness the full power of ASP.NET using any text editor -- even Notepad! But Visual Studio .NET adds the productivity of Visual Basic-style development to the Web. Now you can visually design ASP.NET Web

Forms using familiar drag-drop-double-click techniques, and enjoy full-fledged code support including statement completion and color-coding. VS.NET also provides integrated support for debugging and deploying ASP.NET Web applications.

The Enterprise versions of Visual Studio .NET deliver life-cycle features to help organizations plan, analyze, design, build, test, and coordinate teams that develop ASP.NET Web applications. These include UML class modeling, database modeling (conceptual, logical, and physical models), testing tools (functional, performance and scalability), and enterprise frameworks and templates, all available within the integrated Visual Studio .NET environment.

- **Rich Class Framework:** Application features that used to be hard to implement, or xrequired a 3rd-party component, can now be added in just a few lines of code using the .NET Framework. The .NET Framework offers over 4500 classes that encapsulate rich functionality like XML, data access, file upload, regular expressions, image generation, performance monitoring and logging, transactions, message queuing, SMTP mail, and much more!

Improved Performance and Scalability

ASP.NET lets you use serve more users with the same hardware.

- **Compiled execution:** ASP.NET is much faster than classic ASP, while preserving the "just hit save" update model of ASP. However, no explicit compile step is required! ASP.NET will automatically detect any changes, dynamically compile the files if needed, and store the compiled results to reuse for subsequent requests. Dynamic compilation ensures that your application is always up to date, and compiled execution

makes it fast. Most applications migrated from classic ASP see a 3x to 5x increase in pages served.

- **Rich output caching:** ASP.NET output caching can dramatically improve the performance and scalability of your application. When output caching is enabled on a page, ASP.NET executes the page just once, and saves the result in memory in addition to sending it to the user. When another user requests the same page, ASP.NET serves the cached result from memory without re-executing the page. Output caching is configurable, and can be used to cache individual regions or an entire page. Output caching can dramatically improve the performance of data-driven pages by eliminating the need to query the database on every request.

- **Web-Farm Session State:** ASP.NET session state lets you share session data user-specific state values across all machines in your Web farm. Now a user can hit different servers in the web farm over multiple requests and still have full access to her session. And since business components created with .NET are free-threaded, you no longer need to worry about thread affinity.

- **.NET Outperforms J2EE:** In a head-to-head comparison of performance and scalability between Sun's Java Pet Store J2EE blueprint application and the ASP.NET implementation, .NET significantly outperformed J2EE. The bottom line: the ASP.NET implementation required only $1/4^{th}$ as many lines of code, was 28x faster (that's 2700%), and supported 7.6x as many concurrent users as J2EE, with only $1/6^{th}$ as much processor utilization.

Enhanced Reliability

ASP.NET ensures that your application is always available to your users.

- **Memory Leak, Deadlock and Crash Protection :** ASP.NET automatically detects and recovers from errors like deadlocks and memory leaks to ensure your application is always available to your users.

For example, say that your application has a small memory leak, and that after a week the leak has tied up a significant percentage of your server's virtual memory. ASP.NET will detect this condition, automatically start up another copy of the ASP.NET worker process, and direct all new requests to the new process. Once the old process has finished processing its pending requests, it is gracefully disposed and the leaked memory is released. Automatically, without administrator intervention or any interruption of service, ASP.NET has recovered from the error.

Easy Deployment

ASP.NET takes the pain out of deploying server applications.

- **"No touch" application deployment:** ASP.NET dramatically simplifies installation of your application. With ASP.NET, you can deploy an entire application as easily as an HTML page: just copy it to the server. No need to run regsvr32 to register any components, and configuration settings are stored in an XML file within the application.

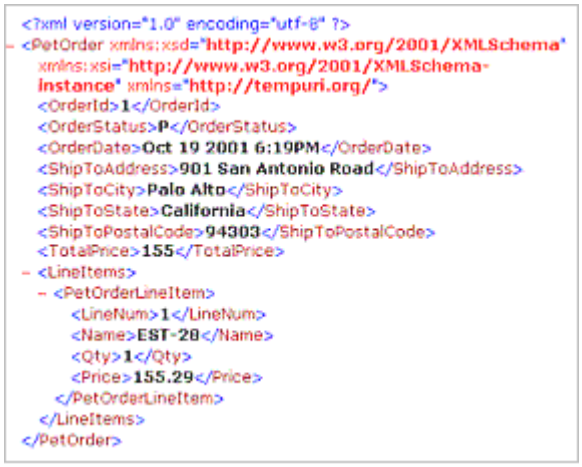
- **Dynamic update of running application:** ASP.NET now lets you update compiled components without restarting the web server. In the past with classic COM components, the developer would have to restart the web server each time he deployed an update. With ASP.NET, you simply copy the component over the existing DLL -- ASP.NET will automatically detect the change and start using the new code.

- **Easy Migration Path:** You don't have to migrate your existing applications to start using ASP.NET. ASP.NET runs on IIS side-by-side with classic ASP on Windows 2000 and Windows XP platforms. Your existing ASP applications continue to be processed by ASP.DLL, while new ASP.NET pages are processed by the new ASP.NET engine. You can migrate application by application, or single pages. And ASP.NET even lets you continue to use your existing classic COM business components.

New Application Models

ASP.NET extend your application's reach to new customers and partners.

- **XML Web Services:** XML Web services allow applications to communicate and share data over the Internet, regardless of operating system or programming language. ASP.NET makes exposing and calling XML Web Services simple. Any class can be converted into an XML Web Service with just a few lines of code, and can be called by any SOAP client. Likewise, ASP.NET makes it incredibly easy to call XML Web Services from your application. No knowledge of networking, XML, or SOAP is required.



- **Mobile Web Device Support:** ASP.NET Mobile Controls let you easily target cell phones, PDAs -- over 80 mobile Web devices -- using ASP.NET. You write your application just once, and the mobile controls automatically generate WAP/WML, HTML, or iMode as required by the requesting device.

Because of these reasons I decided to use ASP.NET (C#) for devoloping this project.

ASP

ASP is a server-side scripting technology developed by Microsoft. It is an open, compile-free application environment in which you can combine HTML, scripts, and reusable components to build dynamic and powerful Web applications.

An ASP application consists of ASP pages published on a Web site. ASP pages can contain HTML code, client-side scripts, and server-side scripts. When a user goes to an ASP page, the Web server calls the ASP Server, which processes the requested file from top to bottom, executing any server-side scripts. It then formats a standard Web page and sends the results to the user's browser.

Because scripts can run on the server rather than on the client, the Web server can do much of the work involved in generating the HTML pages sent to browsers. Server-side scripts cannot be readily copied because only the result of the script is returned to the browser. Users cannot view the script commands that created the page they are viewing.

ASP was designed as a faster and easier alternative to CGI scripting using Perl or C scripts. ASP provides an easy-to-learn scripting interface (including native support for both VBScript and JavaScript), along with a number of predefined objects that simplify many development tasks, such as maintaining user state and defining global variables within an application. You can also use Active-X Data Objects (ADO): components to perform additional functions, among them accessing ODBC-complaint databases and outputting data to text files.

You can extend ASP scripts by using component object model (COM) components, Java components, and extensible markup language (XML).

HTML and Static Web Pages

Initially, most Web sites were used to publish information. Typical content included items like company profiles, press releases, annual reports, product brochures, etc. HTML tags were used to tell the user's Web browser (the client side) how to display static information stored on a computer in HTML files.

For example, if a user types a URL, the Web browser looks for the Web site, finds the Web server on the Internet, and requests that the page be sent to the Web browser. After the Web server finds the page, it sends it to the Web browser. The Web browser reads the HTML tags in the page and converts them into a formatted page. In this respect, HTML is a display language that displays static pages. There is not a way for the user to send information back to the Web server.

Scripting Languages and Dynamic Web Pages

In order for Web sites to be dynamic and gather information from the user, developers needed a more powerful tool, one that enabled them to write code alongside HTML and have that code executed when a page is requested. Client-side scripting, such as VBScript or JavaScript, is such a tool. It can be used to do simple interactive tasks such as verifying that a user had entered all the necessary fields prior to sending an order to a Web server. The browser executes client-side scripts after the page has been received from the server. While client-side scripts enable the dynamic flow of information between the browser and the server, they have two main weaknesses:

- The source code can be viewed from a user's browser and could be stolen.
- client-side scripts can only run on browsers that support the scripting language the developer used.

CGI and Interactive Web Sites

Then an alternative called server-side scripting emerged. CGI (Common Gateway Interface) is a computer program written in Perl or C that enables the Web server to talk to other applications before displaying a file in a user's browser. CGI translates the information coming from the user for the Web server, then saves it to a text file or database. Most server-side tools like CGI must be compiled and associated with the appropriate Web pages. While interactive, there are several limitations associated with CGI:

- Requires Web developers with knowledge of Perl or C.
- Results in inefficient processing since every new request to a CGI creates a new process on the server and opens a new copy of the CGI application. Having multiple copies running concurrently can result in lengthy response times, especially to high traffic sites.
- Leads to longer development times since CGI developers cannot take advantage of visual development interfaces like Visual Basic.
- Raises security concerns in that by running on the server side, CGI opens up your computer to potential fraud.
- Not all Internet service providers (ISPs) host CGI scripts because they don't want to take the risk of a poorly written CGI code crashing the server and every site hosted on that server, or risk performance problems.

APIs and Interactive Web Sites

Another server-side scripting tool, APIs (Application Programming Interfaces) is in the mix as well. Like CGI, APIs allow the Web server to communicate with other applications running on the server machine, but they do so without the inefficient processing risks. However, APIs still require complex interfaces to the Web server and are difficult to create

and maintain. In the final analysis, few organizations can afford the time and expense associated with using highly skilled developers to create engaging Web sites with CGI or APIs.

3.1.1.4. Operating System

It is planning to generate this project on Windows 2000 professional operating system.

3.1.2. Determining the Hardware Tools To Be Used In the System

The company where this shopping side has been developed for must have the system that has at least minimum system requirements because otherwise the system will run properly. Existence hardware has enough capacity to provide requirements of project. So, it is not necessary to use an extra harware. The computer is used in the company has these features;

Pentium 4 CPU 3.00 GHz

512 MB RAM

80 GB hard disc

56 K fax-modem

Microsoft Windows 2000 pro operating system

Microsoft Office 2000 office tool

3.2. Economical Feasibility

At this part I calculated the total cost of the system as hardware and software view. An other item of total cost is development cost. Extra cost is not necessary. Because Internet Information Server, SQL Server 2000 and Windows 2000 that are needed to carry out project are present. Besides, it is necessary to benefit from web hosting services so as to broadcast web pages. Companies that provide web hosting service demand about 130$ for a year.

If people who work for project take 1000$:

1 person*3 months*1000$=3000$

Since present hardware is enough and extra hardware is not required for process of carring out project, there won't be extra cost.

3.3. Functional Feasibility

In functional feasibility I worked on the systems; usefulness and functionality and how to develop this kind of structure, which interfaces will be designed, which of them is makes the system run slower and not easy to use. Because making many interfaces some times causes the system not to be efficient, stable. And also deciding the DB structure and its fields started to be more and more clearer at this point. In the end, I decided to basic structure of the system, which ASP.NET's to develop and which tables and its fields to be designed. The specified user will be able to monitor, update, delete the stock, stock items, prices and other administrator tasks. They will be able to use some web based system and they will not need some extra stock monitor programs. On the other hand customers will be able to easily choose the items they want to buy. The system that will be carried out will be accepted by users easily. Beause, users can be adapted easily. Besides, an internet connection is enough to use system. In addition to this, it is not compulsory to work in office.

3.4. Legal Feasibility

This project will be developed by Haliç University Computer Engineering Department. All rights are reserved to the department. Any or all part of this program can't be reproduced and used in other programs without written permission of the department.

4. REQUIREMENT ANALYSIS

The scope of information system was determined, after meeting the association. System will be carried out stage by stage. These stages are;

4.1. Requirement Analysis Stages

- Designing project and having ideas of authority in people in association.
- Designing reports forming project.
- Designing tables forming project.
- Designing forms forming project.
- Preparing program modules forming project.
- Relating among table, form, report and program modules.
- Testing project.
- Showing project to authority in people in association and having their ideas.
- Preparing documents.

4 . 2 . Determining Problems That Can Reveal In Requirement Analysis And Solution Methods

All elements and components are examined in detail and added informations are modulated and indicated with diagrams in requirement analysis. Risk Management which is the most important stage support us is carried out in project planning. In risk management, the factors that can be risk are evaluated. This study is a precaution against the problems that can reveal in system implementation. Probable problems that can appear are these;

- **Determining requirements of system correctly :** Because enough knowledges about system aren't gathered during meetings, requirements of system can't be determined

accurately. If we face problem like that, we have to return the beginning of project and determine requirements again.

- **Users may not be enough qualified for using :** Shopping program that will form should be user friendly. Because, users may not have enough knowledge to use. So, the use of project should be understandable for everybody.
- **The system may not be described enough :** Program should be explained accurately to users. Because, if users don't have enough knowledges about functions of program, they can not benefit from program accurately. Knowledges about program should be given to users not to face problem like that.
- **The lack of documents :** Studies, results of test and diagrams should be documented accurately. If documents are not prepared well, it causes unintelligible in system during design, implementation and maintain stages. For this reason, documents about studies should be prepared for every stage.
- **Program should be flexible :** In case new requirements appear later, new menus and modules will be able to be added. Thus, writing the most of codes can be prevented.
- **Testing program :** In program implementation stage, while passing one step to another step, it is necessary to complete test process. If it is not, problems can appear. So, at the end of every stage, program have to be tested according to current stage.
- **Writing understandable codes :** In system implementation, explanations should be found in algorithms and codes. In program maintain stage or when the faults appear, we have to return the beginning of program to revise codes. So, if codes of program are understandable, it facilitates our studies.

5 . SYSTEM DESIGN

In this part, the system is designed as suitable as the properties that are planned in the previous parts.

5 . 1 . Table Structures

In this part, you can see the tables that are used in the project.

Field Name	Data Type
MODID	AutoNumber
MODULSAYFASI	Memo
MODULADI	Text
PARAMETRE	Number
POPUPSAYFASI	Text

Table 1 ANASAYFAMODULLER

Field Name	Data Type
CEVAPID	AutoNumber
ANKETID	Number
CEVAP	Text
BILET	Number
CEVAPTARIHI	Date/Time

Table 2 ANKETCEVAP

Field Name	Data Type
ANKETID	AutoNumber
ANKETADI	Text
SORU	Text
SECSAYISI	Number
SEC1	Text
SEC2	Text
SEC3	Text
SEC4	Text
SEC5	Text
SEC6	Text
SEC7	Text
SEC8	Text
SEC9	Text
SEC10	Text
DIGERKULLAN	Number
KAYITTARIH	Date/Time
GUNCELLEMETARIHI	Date/Time
BASLANGICTARIHI	Date/Time
BITISTARIHI	Date/Time
UYELEREOZEL	Number
BIRKULLANIM	Number
AKTIF	Number
ONSAYFA	Number
EKLEYEN	Number

Table 3 ANKETLER

	Field Name	Data Type
🔑	AYARLARID	Number
	BOLGESAYISI	Number
	ANASAYFABOLGESAYISI	Number
	DUYURUSAYFASIID	Number
	SAYFABASLIGI	Text
	ANASAYFAYEMEKMENUSU	Text

Table 4 AYARLAR

	Field Name	Data Type
🔑	KUTUID	AutoNumber
	BOLGE	Number
	SIRA	Number
	BASLIK	Text
	ICERIK	Memo
	KAYITTARIHI	Date/Time
	GUNCELLEMETARIHI	Date/Time
	EKLEYEN	Number
	GUNCELLEYEN	Number
	AKTIF	Number
	ICON	Text
	SAYFAID	Number
	KUTUTIPI	Number
	MODUL	Number
	MODULID	Number

Table 5 BOLGESISTEM

	Field Name	Data Type
🔑	DUYURUID	AutoNumber
	DUYURUKISAACIKLAMA	Memo
	DUYURUUZUNACIKLAMA	Memo
	DUYURUBASLIK	Text
	KUCUKIMAGEURL	Text
	KAYITTARIHI	Date/Time
	GUNCELLEMETARIHI	Date/Time
	EKLEYEN	Number
	AKTIFMI	Number
	BUYUKIMAGEURL	Text
	BASLANGICTARIHI	Date/Time
	BITISTARIHI	Date/Time

Table 6 DUYURULAR

	Field Name	Data Type
🔑	SORUID	AutoNumber
	SORU	Memo
	CEVAP	Memo
	CEVAPLAYAN	Number
	CEVAPISTENDIMI	Number
	SORUGIRISTARIHI	Date/Time
	SORUCEVAPTARIHI	Date/Time
	KAYITTARIHI	Date/Time
	SORUGUNCELLEMETARIHI	Date/Time
	CEVAPGUNCELLEMETARIHI	Date/Time
	BILETID	Text
	AKTIF	Number
	SORUOKUMA	Number
	CEVAPOKUMA	Number

Table 7 EDANISMA

	Field Name	Data Type
🔑	BILETID	Text
	SORUSAHIBI	Number
	MUDURLUK	Number
	AKTIF	Number
	KAYITTARIHI	Date/Time

Table 8 EDANISMABILET

	Field Name	Data Type
🔑	SAYFAID	AutoNumber
	ANAID	Number
	SIRA	Number
	ICERIKLINK	Text
	HEDEF	Text
	BASLIK	Text
	KAYITTARIHI	Date/Time
	GUNCELLEMETARIHI	Date/Time
	EKLEYEN	Number
	MENURESMI	Text
	ANAHTARKELIMELER	Text
	ACIKLAMA	Text
	AKTIF	Number
	GUNCELLEYEN	Number
	MODUL	Number
	MODID	Number

Table 9 ICERIK

	Field Name	Data Type
🔑	MODULID	AutoNumber
	MODULISMI	Text
	MODULSAYFASI	Text
	UYELIK	Number
	MODULKISAADI	Text

Table 10 ICERIKMODUL

	Field Name	Data Type
🔑	IZINKODU	Text
	ACIKLAMA	Text
	SIRA	Number
	AKTIF	Number

Table 11 IZINKODLARI

	Field Name	Data Type
🔑	ID	AutoNumber
	KULID	Number
	YETKI	Text

Table 12 KULLANICIIZIN

	Field Name	Data Type
🔑	KULID	AutoNumber
	KULLANICIADI	Text
	SIFRE	Text
	AD	Text
	SOYAD	Text
	MUDURLUKID	Number
	EPOSTA	Text
	TELEFON	Text
	FAX	Text
	DAHILI	Text
	MOBILTEL	Text
	SONGIRISTARIHI	Date/Time
	GIRISSAYISI	Number
	EKLEYEN	Number
	KAYITTARIHI	Date/Time
	GUNCELLEMETARIHI	Date/Time
	AKTIF	Number

Table 13 KULLANICILAR

	Field Name	Data Type
🔑	KUTUID	AutoNumber
	BOLGE	Number
	SIRA	Number
	BASLIK	Text
	ICERIK	Memo
	DETAYLINK	Text
	DETAYLINKURL	Text
	KAYITTARIHI	Date/Time
	GUNCELLEMETARIHI	Date/Time
	EKLEYEN	Number
	GUNCELLEYEN	Number
	AKTIF	Number
	HEDEF	Text
	ICON	Text
	KUTUTIPI	Number

Table 14 KUTUSISTEM

	Field Name	Data Type
🔑	KUTUTIPI	Number
	KUTUTIPIADI	Text
	KAYITTARIHI	Date/Time
	KUTUTIPIACIKLAMA	Text
	KUTUTIPIACIKLAMALIADI	Text

Table 15 KUTUTIPLERI

	Field Name	Data Type
🔑	RESIMID	AutoNumber
	RESIMADI	Text
	RESIMACIKLAMA	Text
	RESIMURL	Text
	KATEGORIID	Number
	AKTIF	Number
	RESIMNOTU	Number
	OYSAYISI	Number

Table 16 RESIMGALERISI

	Field Name	Data Type
🔑	RGKATEGORIID	AutoNumber
	RGKATEGORI_ADI	Text
	REKATEGORI_RESIMSAYISI	Number

Table 17 RGKATEGORI

	Field Name	Data Type
🔑	ID	AutoNumber
	EPOSTA	Text
	OKUNDU	Number
	MAILID	Number

Table 18 MAILADRES

	Field Name	Data Type
🔑	GRUPID	AutoNumber
	GRUPADI	Text

Table 19 MAILGRUP

	Field Name	Data Type
🔑	ID	AutoNumber
	GRUPID	Number
	MAILID	Number

Table 20 MAILGRUPUYELERI

	Field Name	Data Type
🔑	MAILID	AutoNumber
	KONU	Text
	GONDEREN	Number
	SABLONID	Number

Table 21 MAILKONU

	Field Name	Data Type
🔑	ID	AutoNumber
	EPOSTA	Text
	AKTIF	Number

Table 22 MAILLIST

	Field Name	Data Type
🔑	SABLONID	AutoNumber
	ADI	Text
	SABLON	Memo

Table 23 MAILSABLON

	Field Name	Data Type
🔑	DEGID	AutoNumber
	MAKALEID	Number
	PUAN	Number
	YORUM	Memo
	TARIHI	Date/Time

Table 24 MAKALEDEG

	Field Name	Data Type
🔑	MAKALEKATEGORI	AutoNumber
	KATEGORIADI	Text
	KAYITTARIHI	Date/Time
	EKLEYEN	Number
	AKTIF	Number
	ANAID	Number
	SIRA	Number

Table 25 MAKALEKATEGORI

	Field Name	Data Type
🔑	MAKALEID	AutoNumber
	BASLIK	Text
	KATEGORIID	Number
	YAZAR	Number
	ANAHTARKELIMELER	Memo
	ACIKLAMA	Memo
	XMLDOSYASI	Text
	AKTIF	Number
	EKLEMETARIHI	Date/Time
	GUNCELLEMETARIHI	Date/Time
	OKUNMASAYISI	Number

Table 26 MAKALELER

	Field Name	Data Type
🔑	MESLEKID	AutoNumber
	MESLEKADI	Text
	AKTIF	Number

Table 27 MESLEK

	Field Name	Data Type
🔑	MUDURLUKID	AutoNumber
	MUDURLUKADI	Text
	MUDURADI	Text
	MUDURSOYADI	Text
	TELEFON	Text
	MOBILTEL	Text
	FAX	Text
	EPOSTA	Text
	GUNCELLEMETARIHI	Date/Time
	EKLEYEN	Number
	KAYITTARIHI	Date/Time
	DEFAULTMUDURLUK	Number

Table 28 MUDURLUK

	Field Name	Data Type
🔑	POPUPID	AutoNumber
	SAYFAID	Number
	ADI	Text
	URL	Text
	GENISLIK	Number
	UZUNLUK	Number
	SOL	Number
	UST	Number
	OLCULENEBILIR	Number
	MENUCUBUK	Number
	KAYANCUBUK	Number
	ALETCUBUK	Number
	YER	Number
	DURUM	Number
	KAPATIRKEN	Text
	ACARKEN	Number
	AKTIF	Number
	KAYITTARIHI	Date/Time
	GUNCELLEMETARIHI	Date/Time
	EKLEYEN	Number
	ICERIK	Memo

Table 29 POPUPSAYFA

	Field Name	Data Type
🔑	ILID	Number
	ILADI	Text

Table 30 SEHIR

	Field Name	Data Type
🔑	SORUID	AutoNumber
	SORU	Memo
	CEVAP	Memo
	KAYITTARIHI	Date/Time
	GUNCELLEMETARIHI	Date/Time
	EKLEYEN	Number
	GOSTER	Number

Table 31 SIKCASORULANLAR

	Field Name	Data Type
🔑	KATID	Number
🔑	SORUID	Number
	SIRA	Number

Table 32 SORUKATEGORI

	Field Name	Data Type
🔑	KATID	AutoNumber
	KATEGORI	Text
	ANAID	Number
	GOSTER	Number
	SIRA	Number

Table 33 SSKATEGORILER

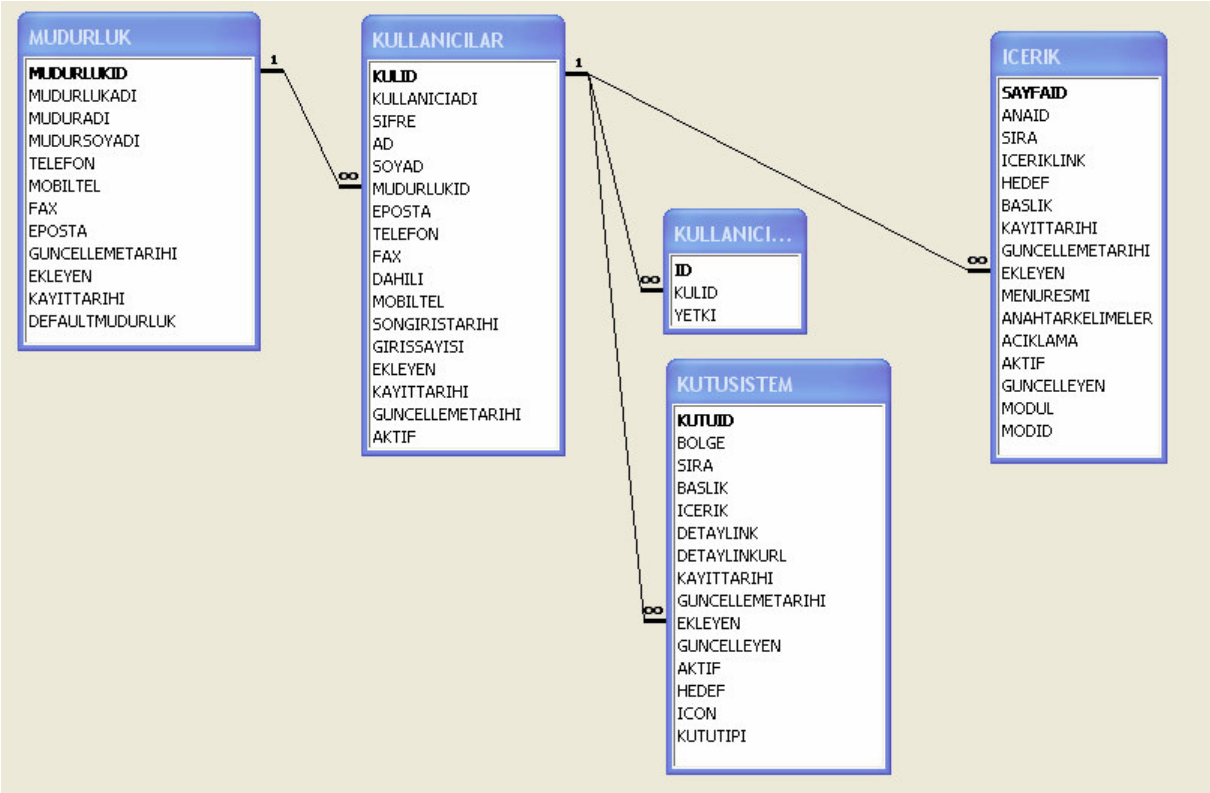
	Field Name	Data Type
🔑	UYEID	AutoNumber
	KULADI	Text
	SIFRE	Text
	SICILNO	Text
	TCKIMLIKNO	Text
	AD	Text
▶	SOYAD	Text
	DOGUMTARIHI	Date/Time
	ADRES	Memo
	MESLEKID	Number
	TELEFON_EV	Text
	TELEFON_IS	Text
	FAX	Text
	MOBILTEL	Text
	EPOSTA	Text
	DOGUMYERI	Number
	ILGIALANLARI	Memo
	GUNCELLEMETARIHI	Date/Time
	KAYITTARIHI	Date/Time
	SONGIRISTARIHI	Date/Time
	GIRISSAYISI	Number
	BEKLENTILER	Memo
	ONAY	Number
	ONAYTARIHI	Date/Time
	AKTIF	Number

Table 34 UYELER

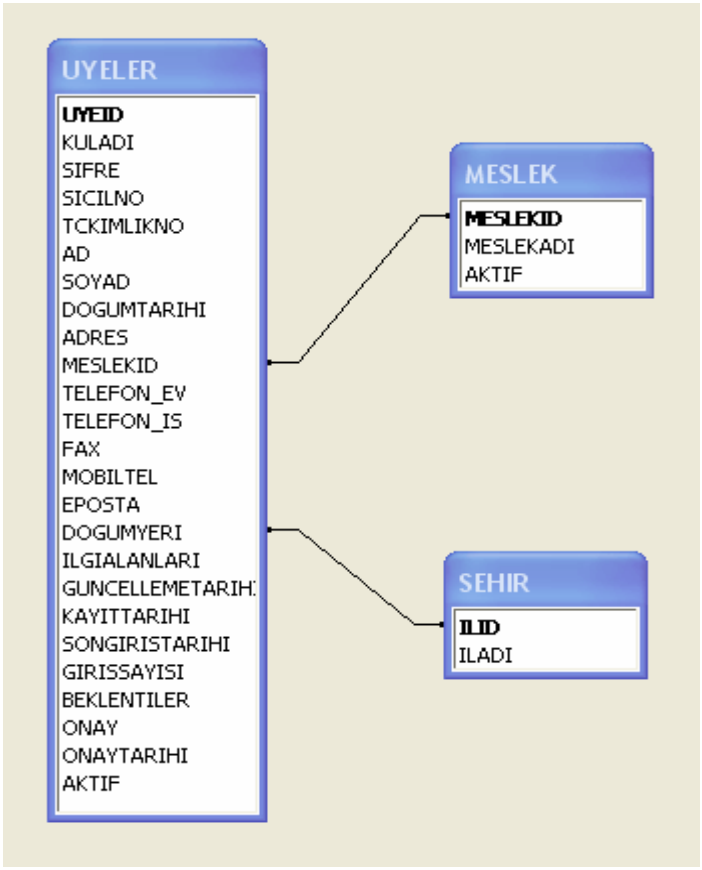
	Field Name	Data Type
🔑	GUN	Number
🔑	AY	Number
🔑	YIL	Number
	YEMEK	Text
	SABAH	Text
	OGLEN	Text
	EKLEYEN	Number
	KAYITTARIHI	Date/Time

Table 35 YEMEKLER

5. 2. Relationships Among Tables

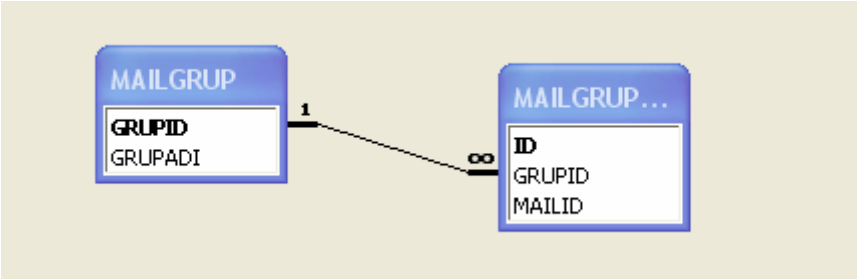


Relationship 1

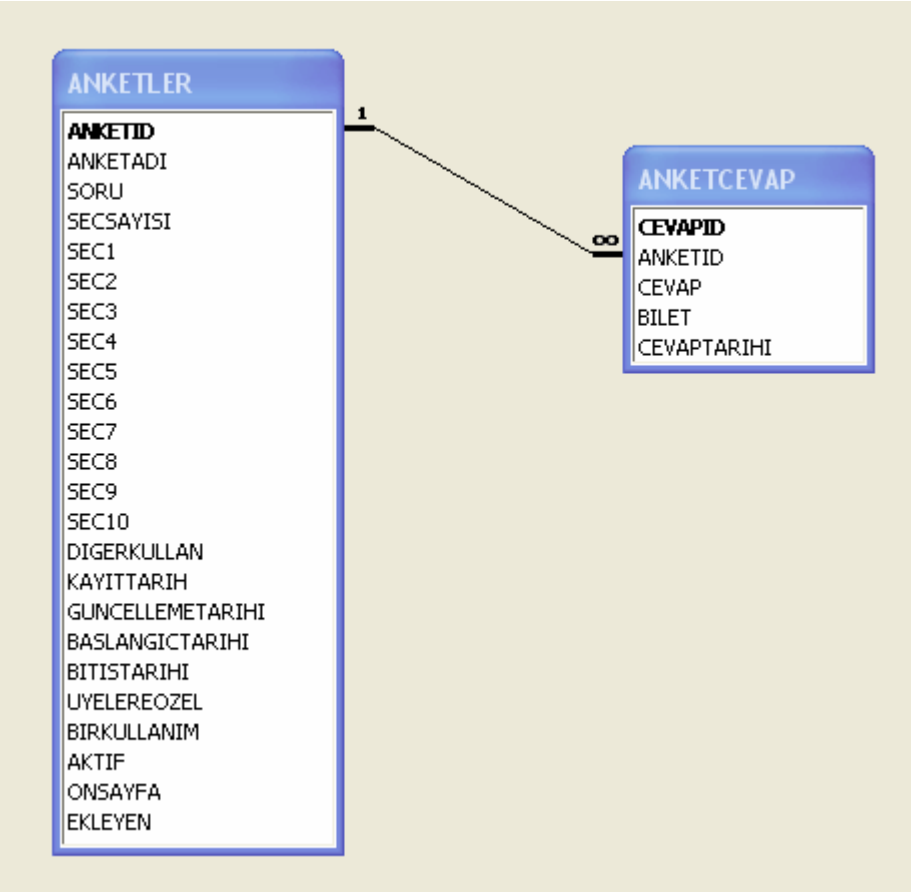


Relationship 2

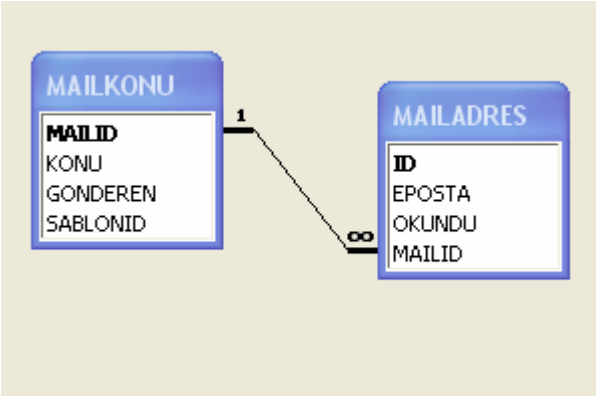
Relationship 3



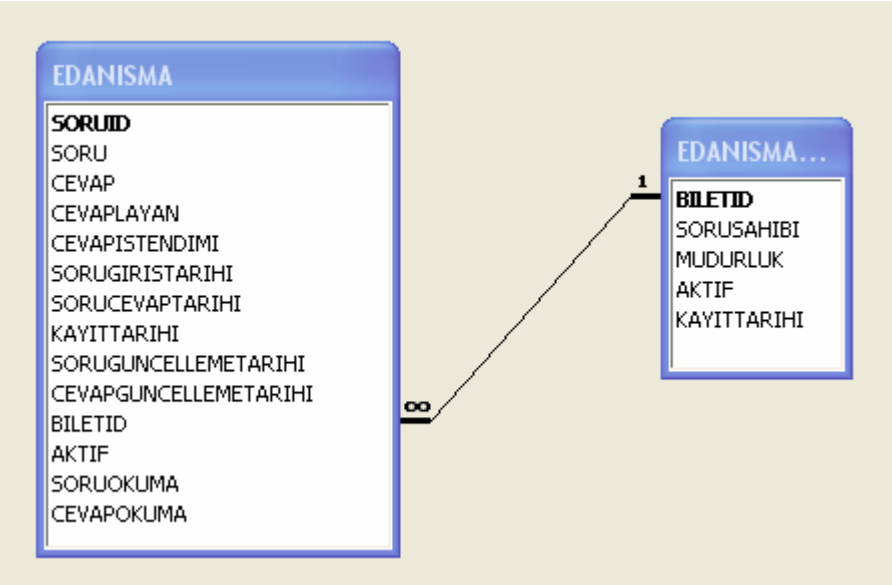
Relationship 4



Relationship 5



Relationship 6



Relationship 7

5. 3. Form Samples

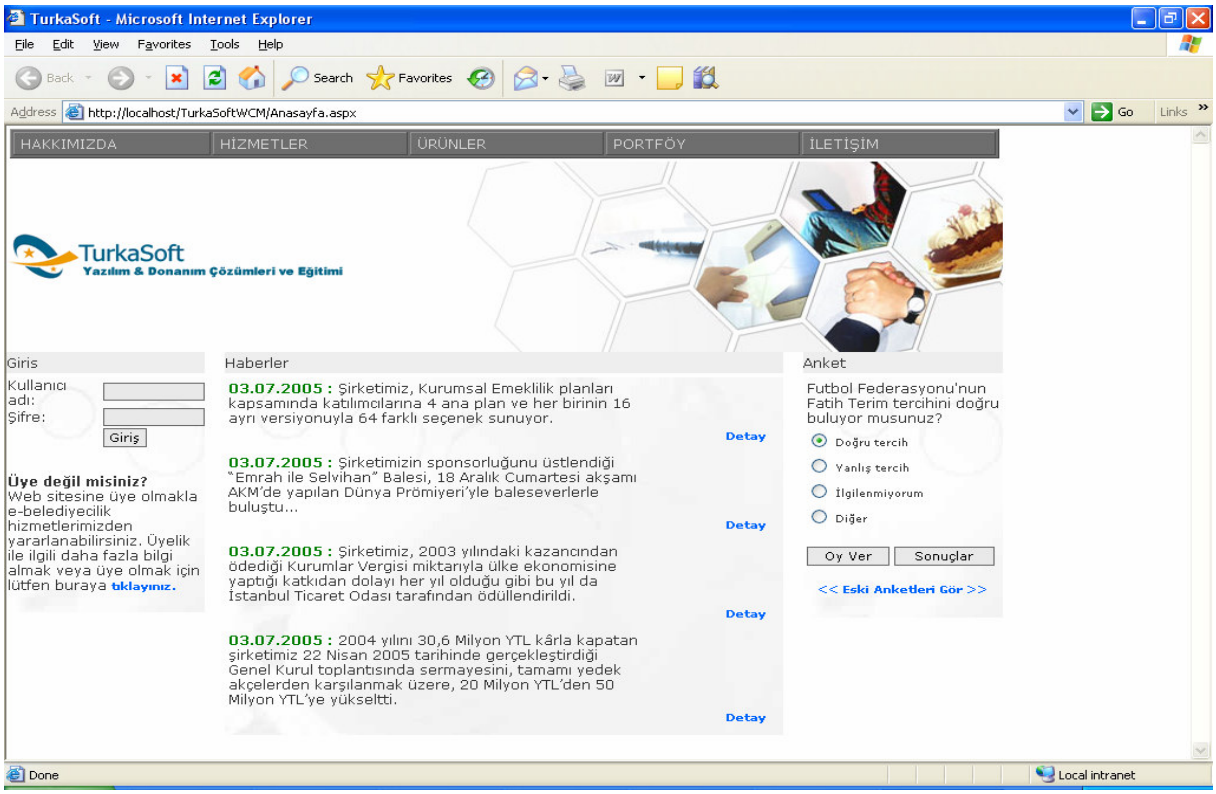


Figure 3 Home Page

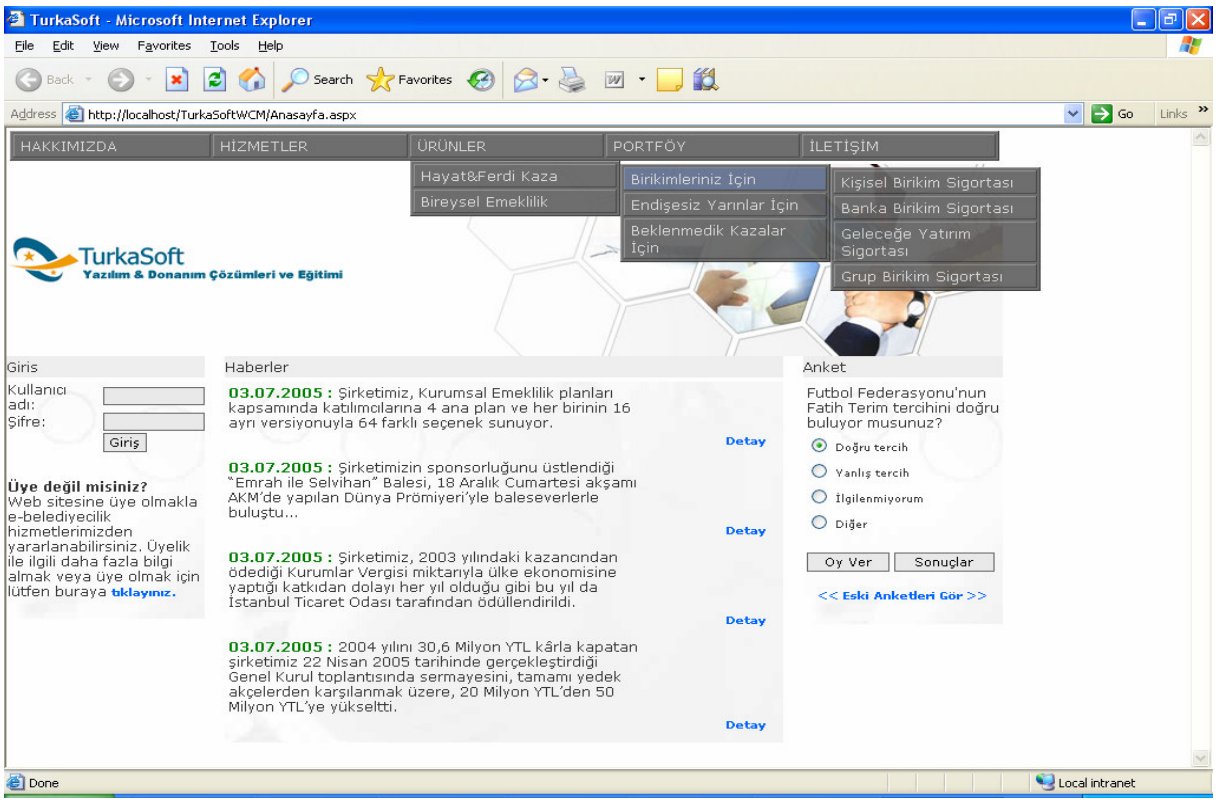


Figure 4 Home Page

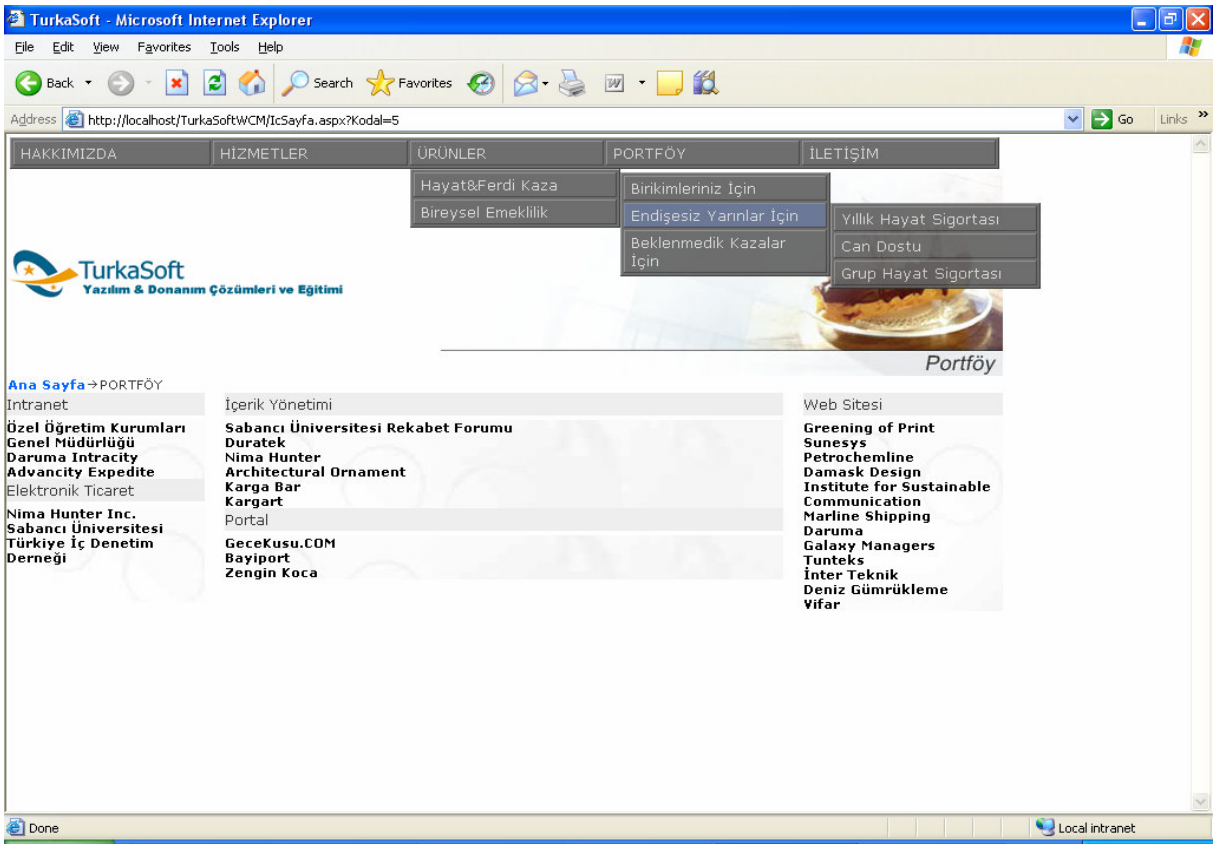


Figure 5 References Page

Figure 6 About Us Page

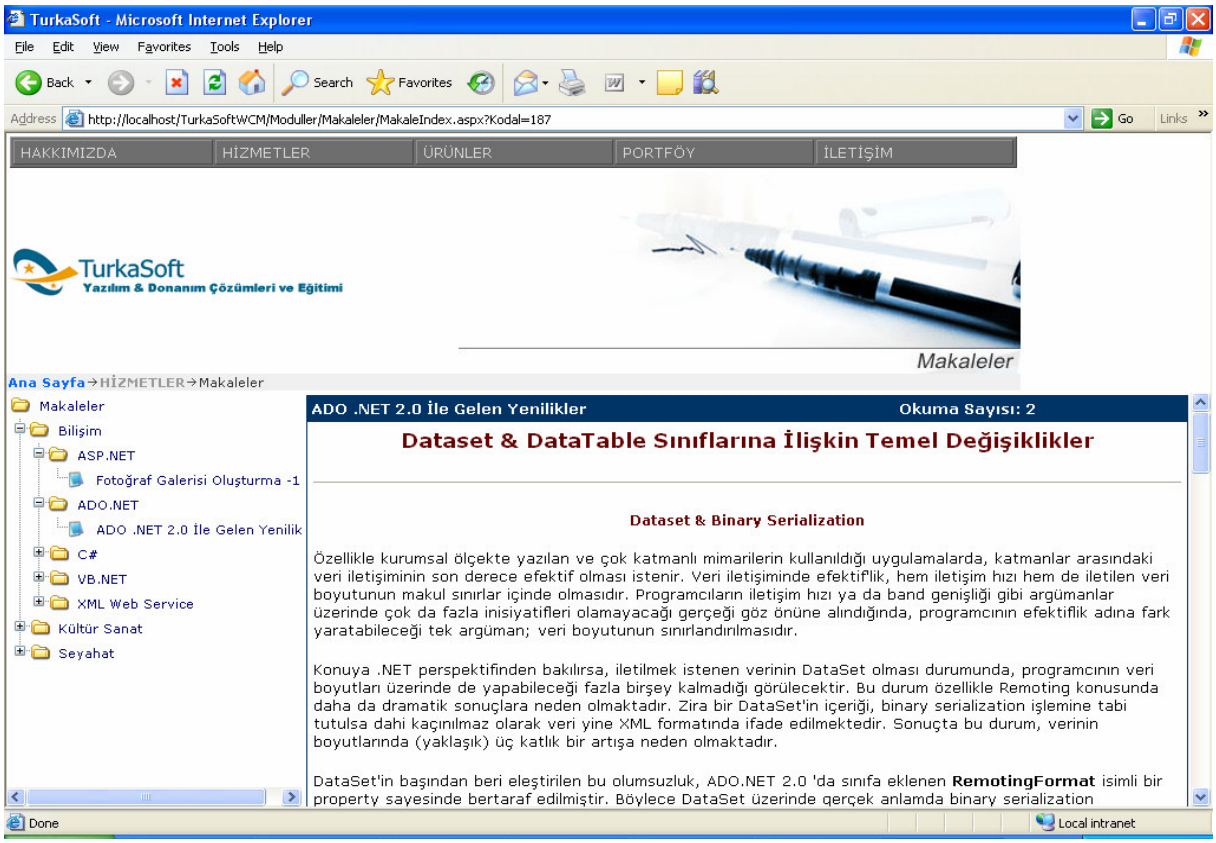


Figure 7 Article Page

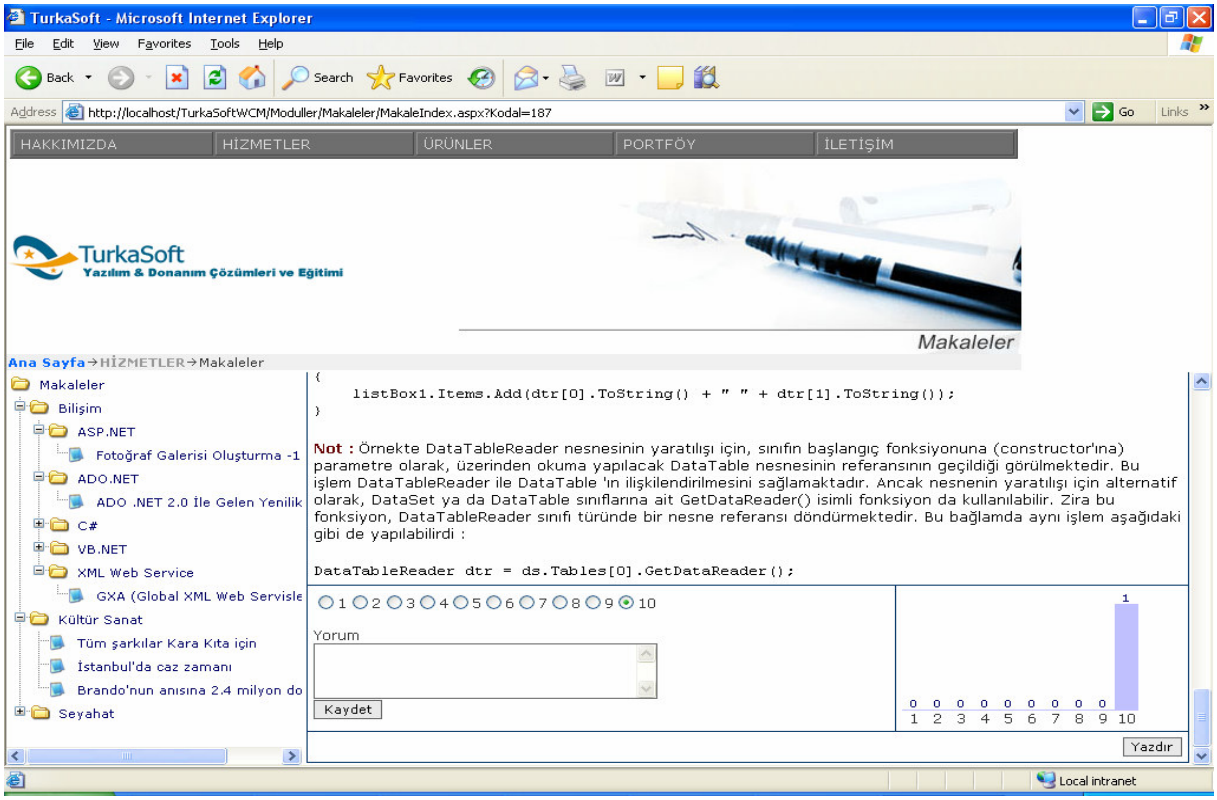


Figure 8 Article Page

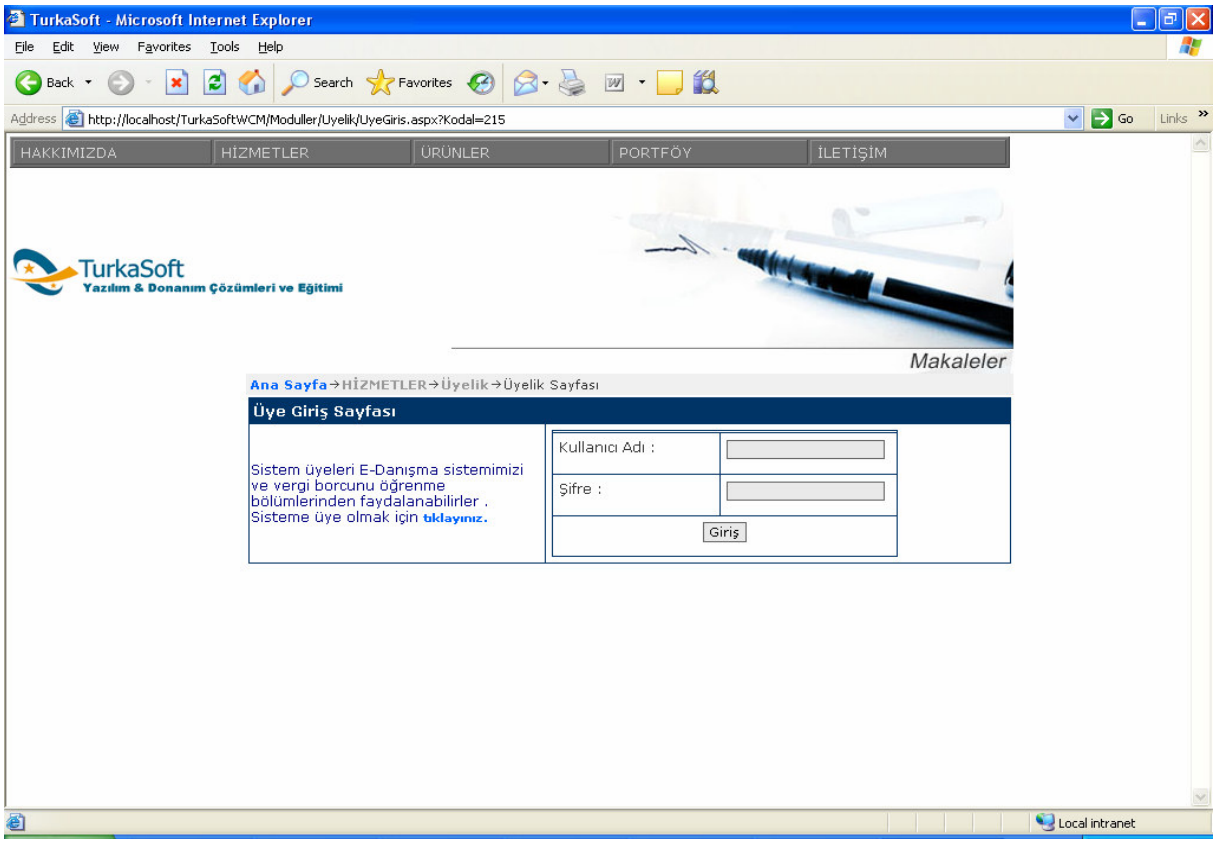


Figure 9 Membership Login Page

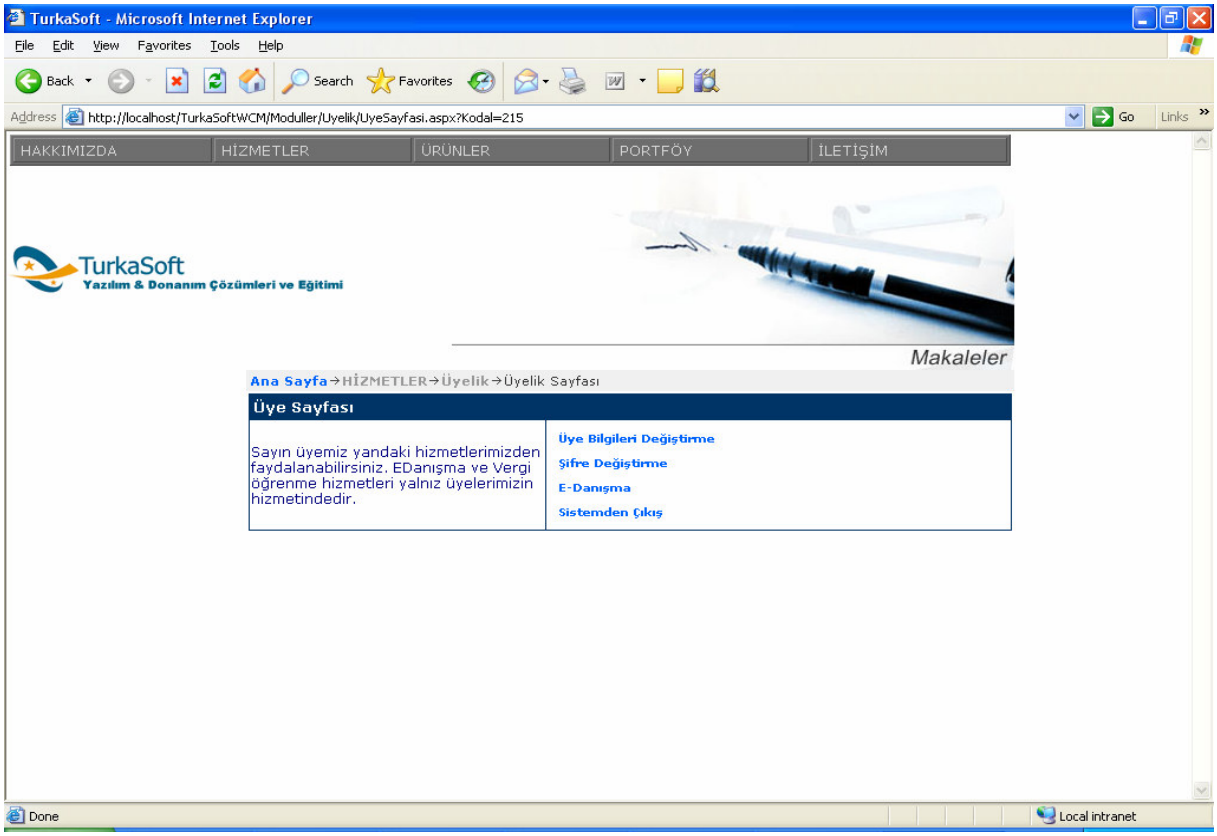


Figure 10 Membership Page

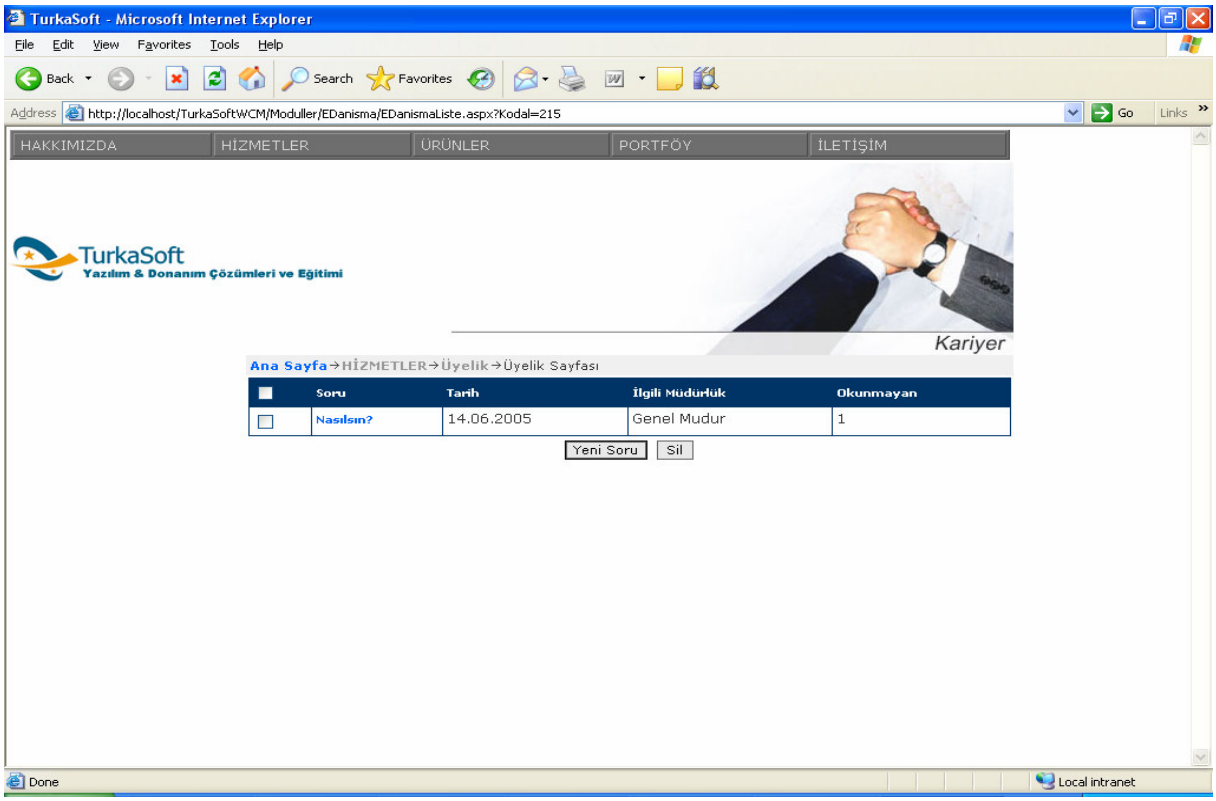


Figure 11 E-Support Page

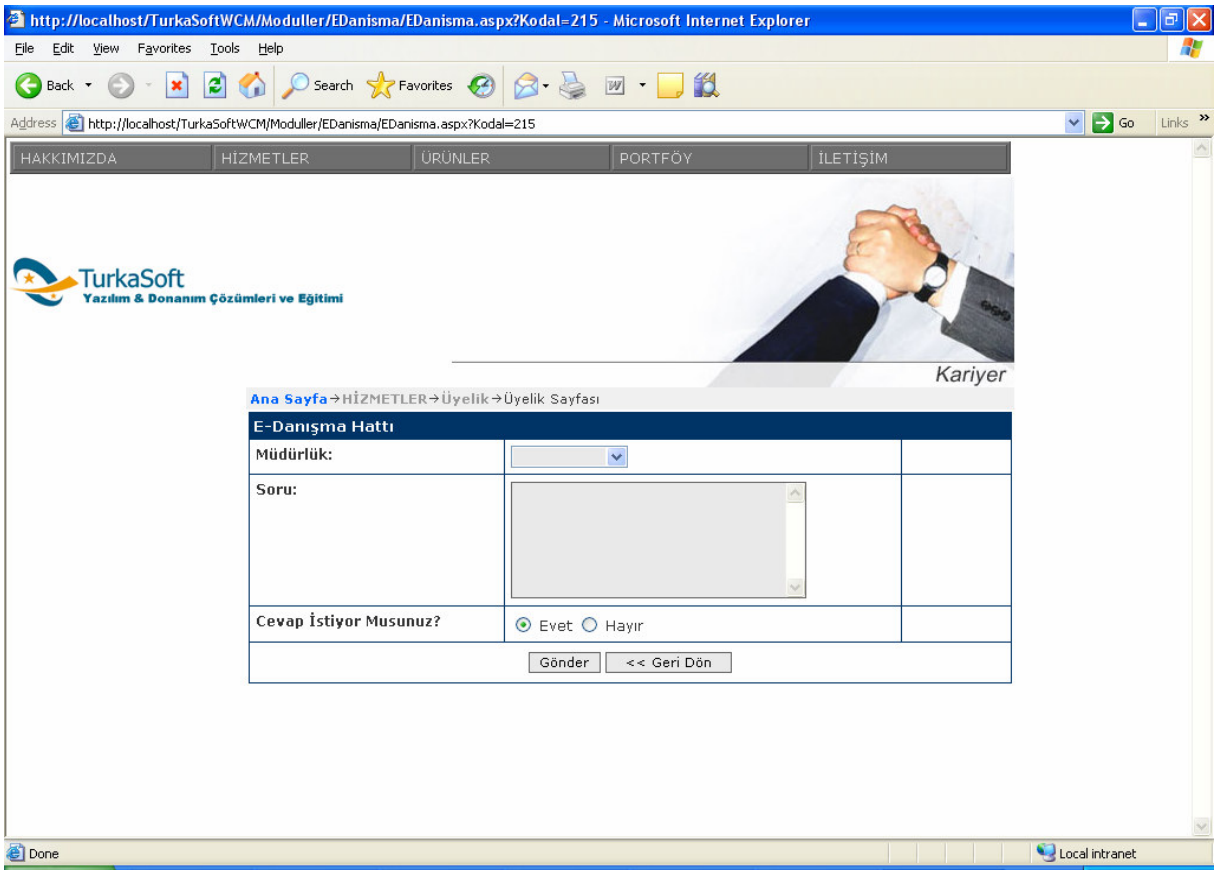


Figure 12 E-Support New Record Page

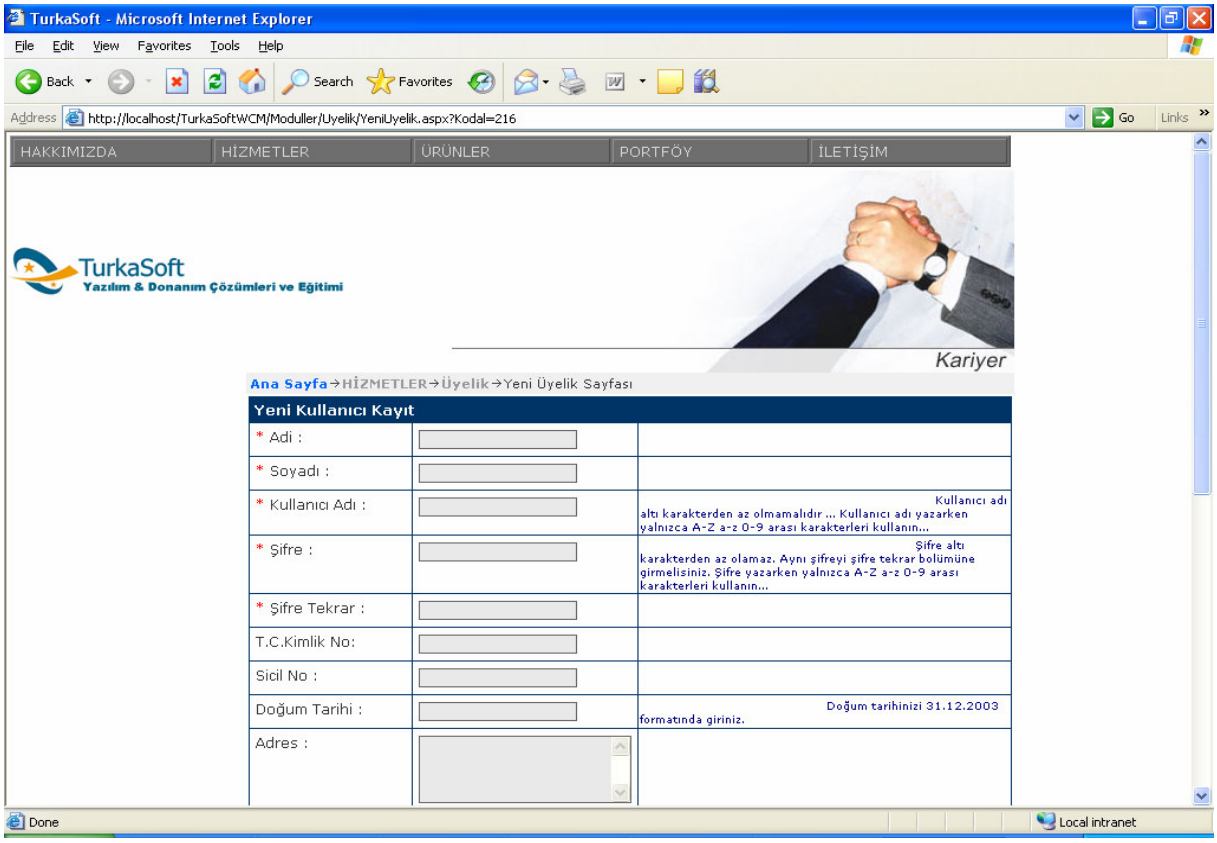


Figure 13 Membership New Record Page

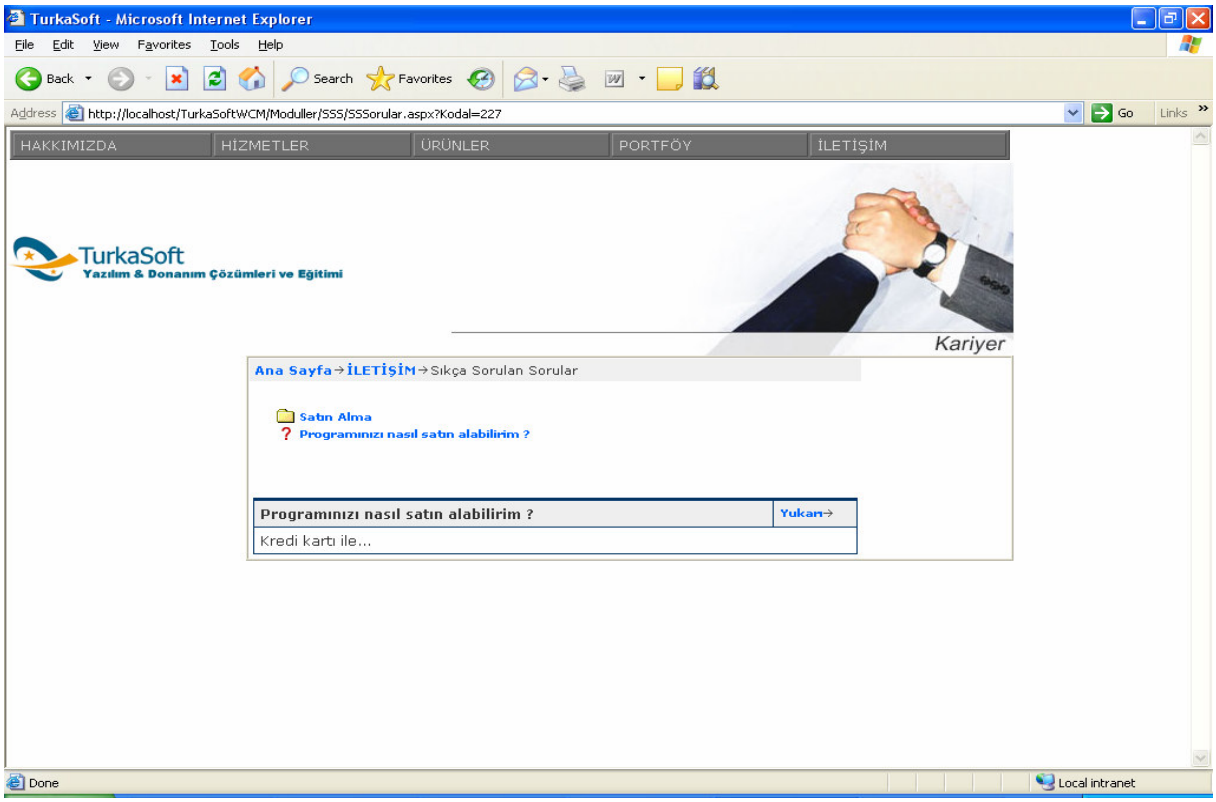


Figure 14 FAQ Page

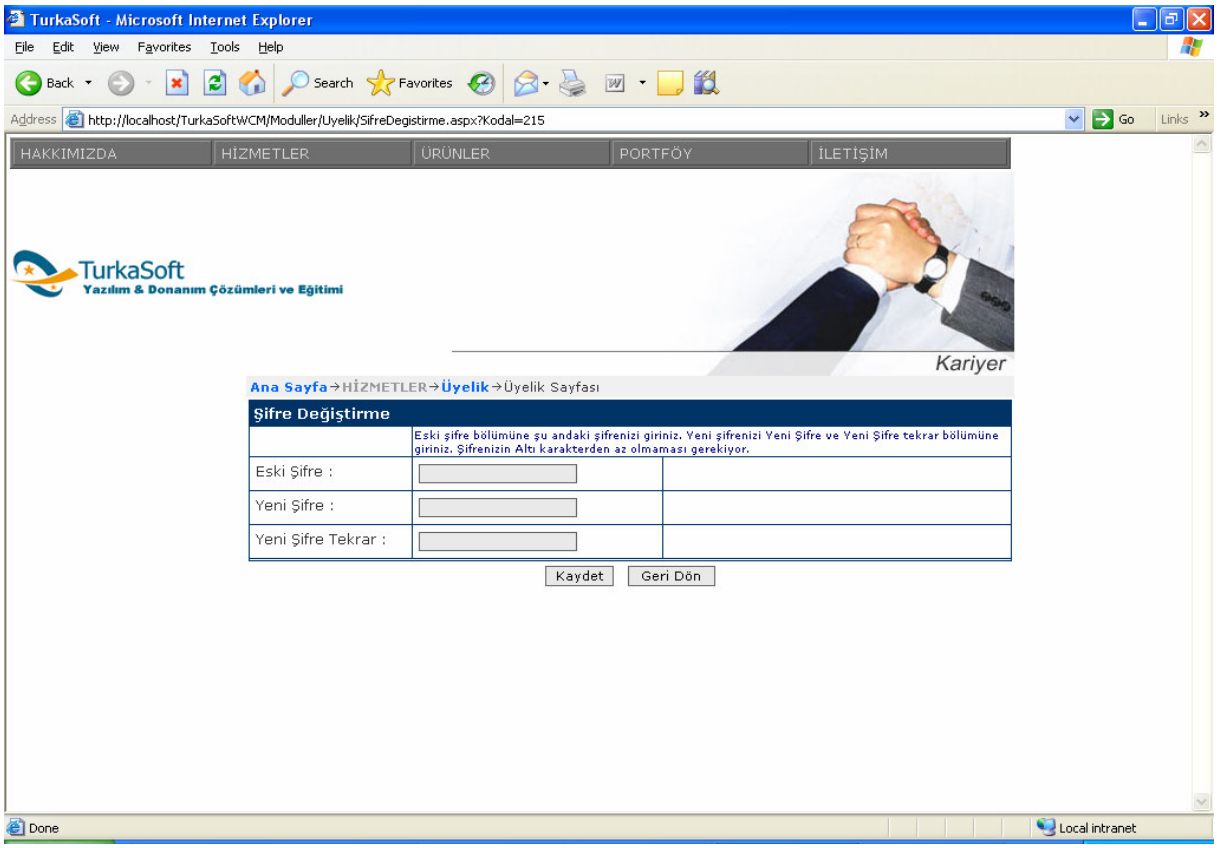


Figure 15 Membership Password Change Page

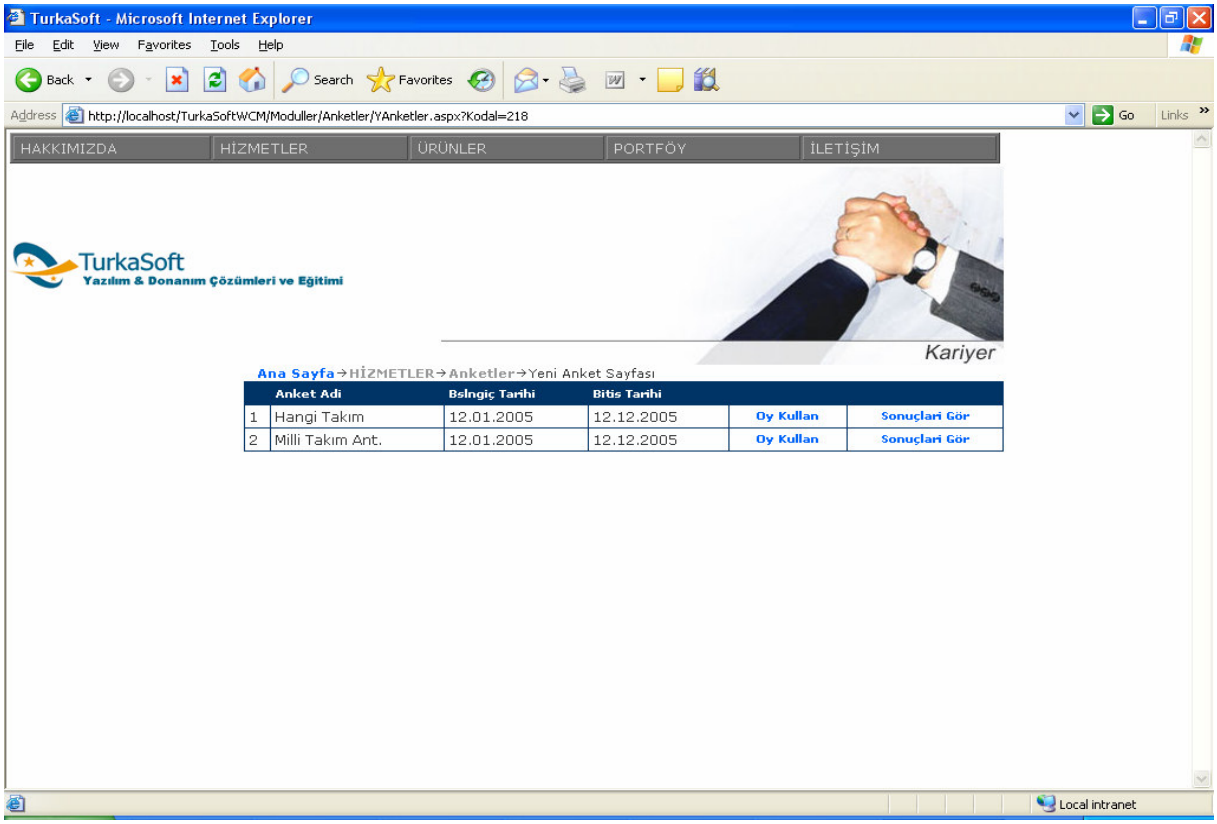


Figure 16 Poll List Page

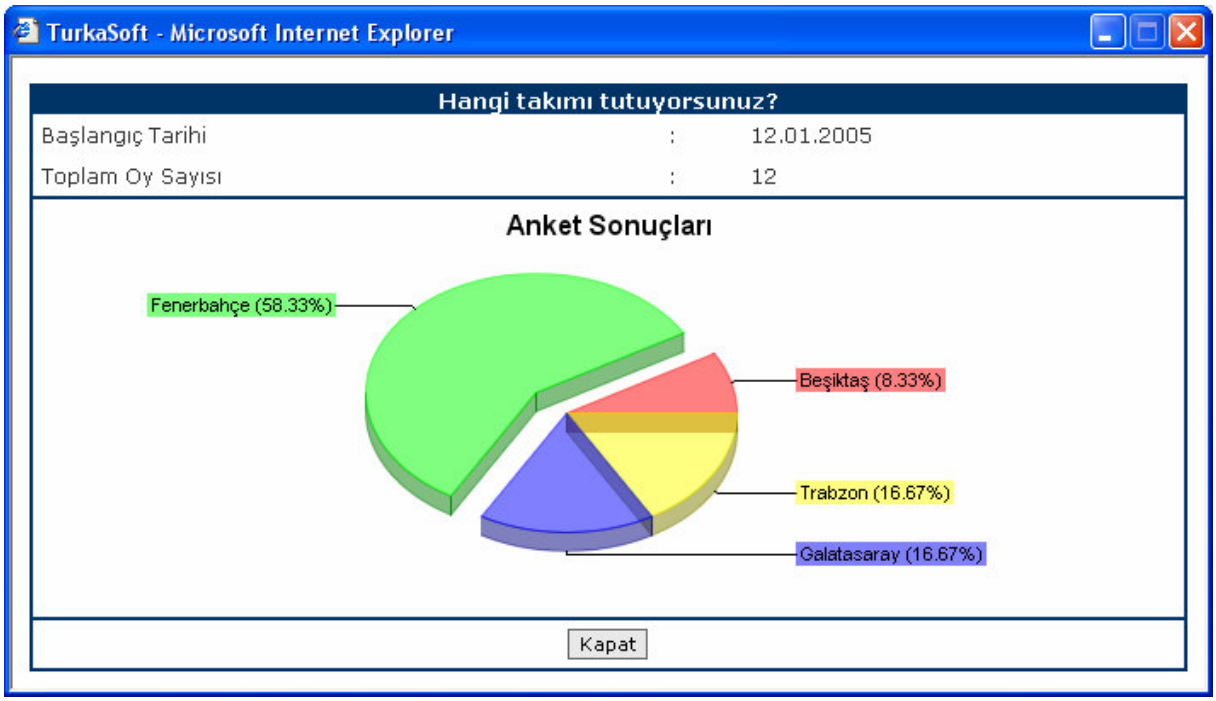


Figure 17 Poll Result Page

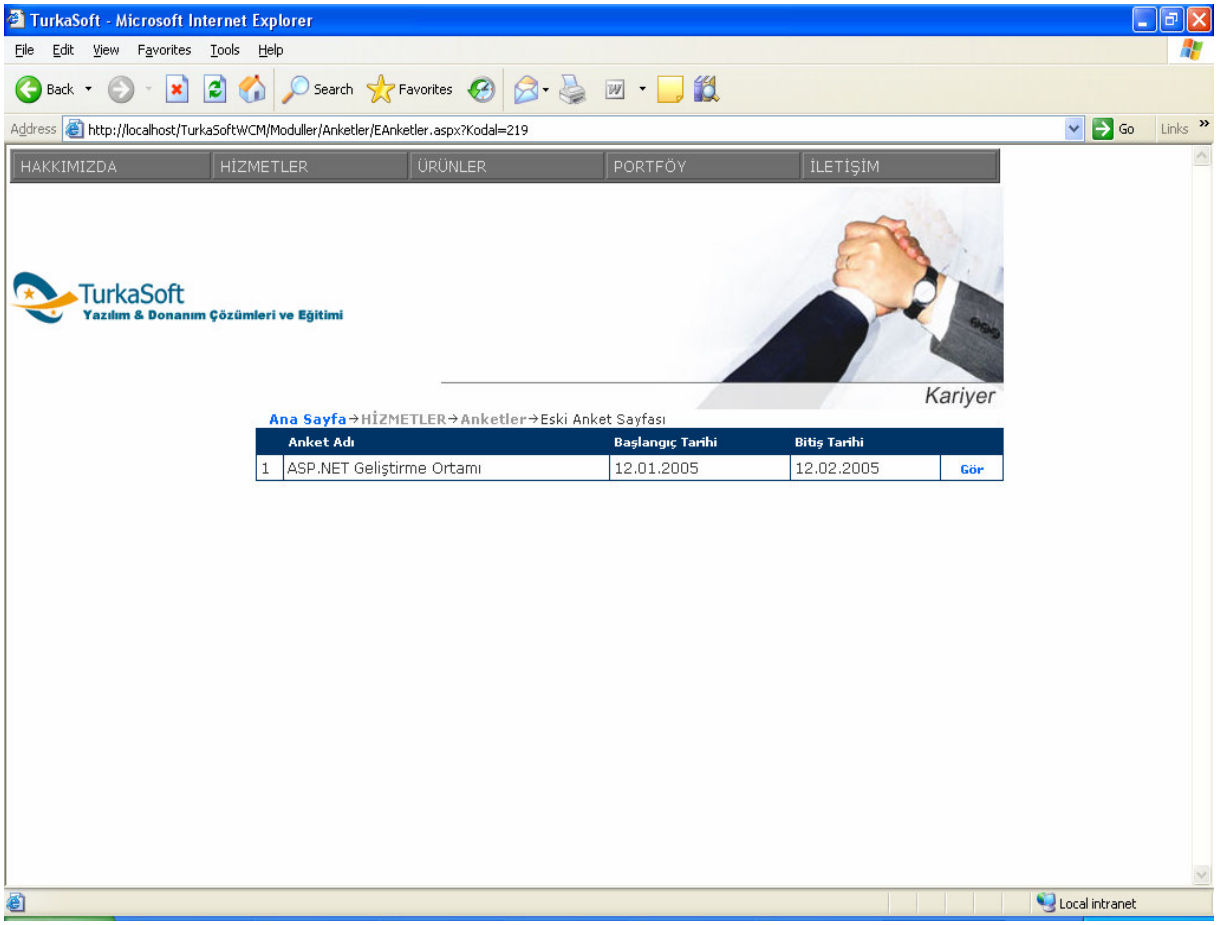


Figure 18 Old Poll List Page

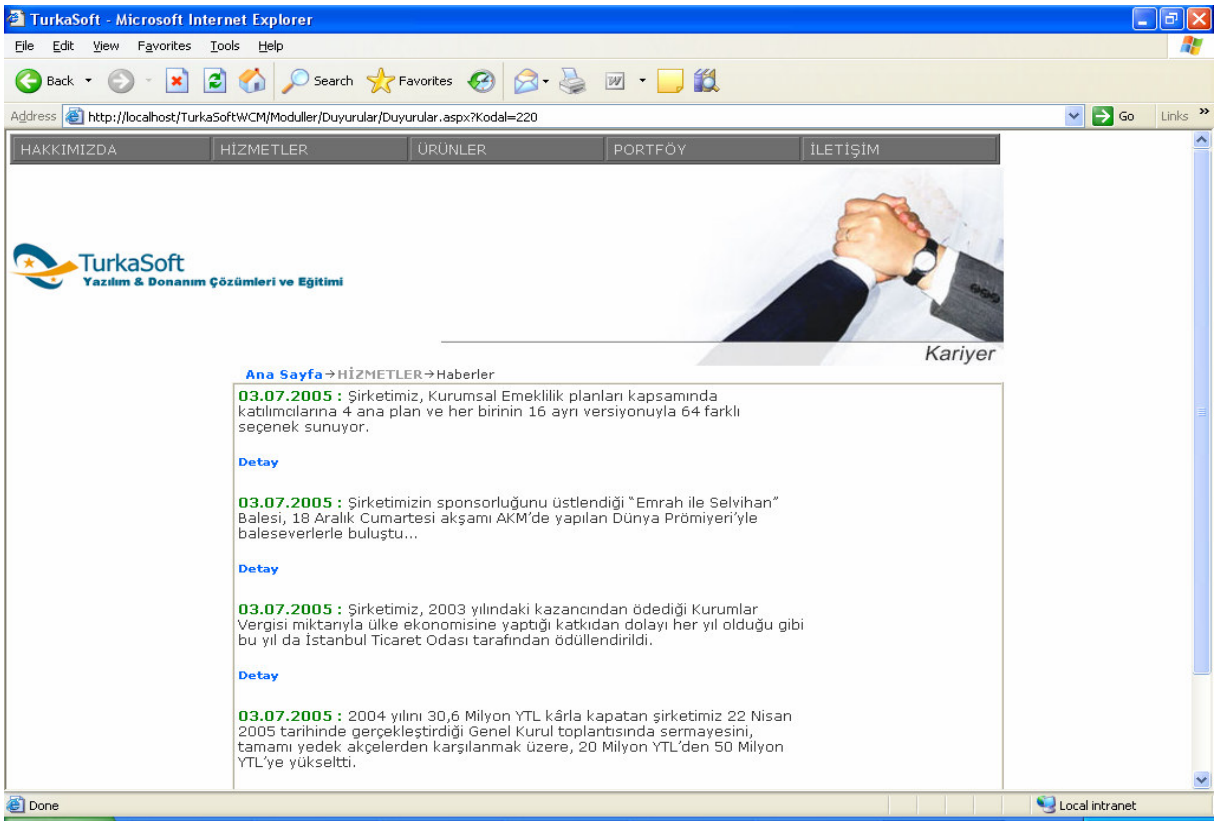


Figure 19 News Page

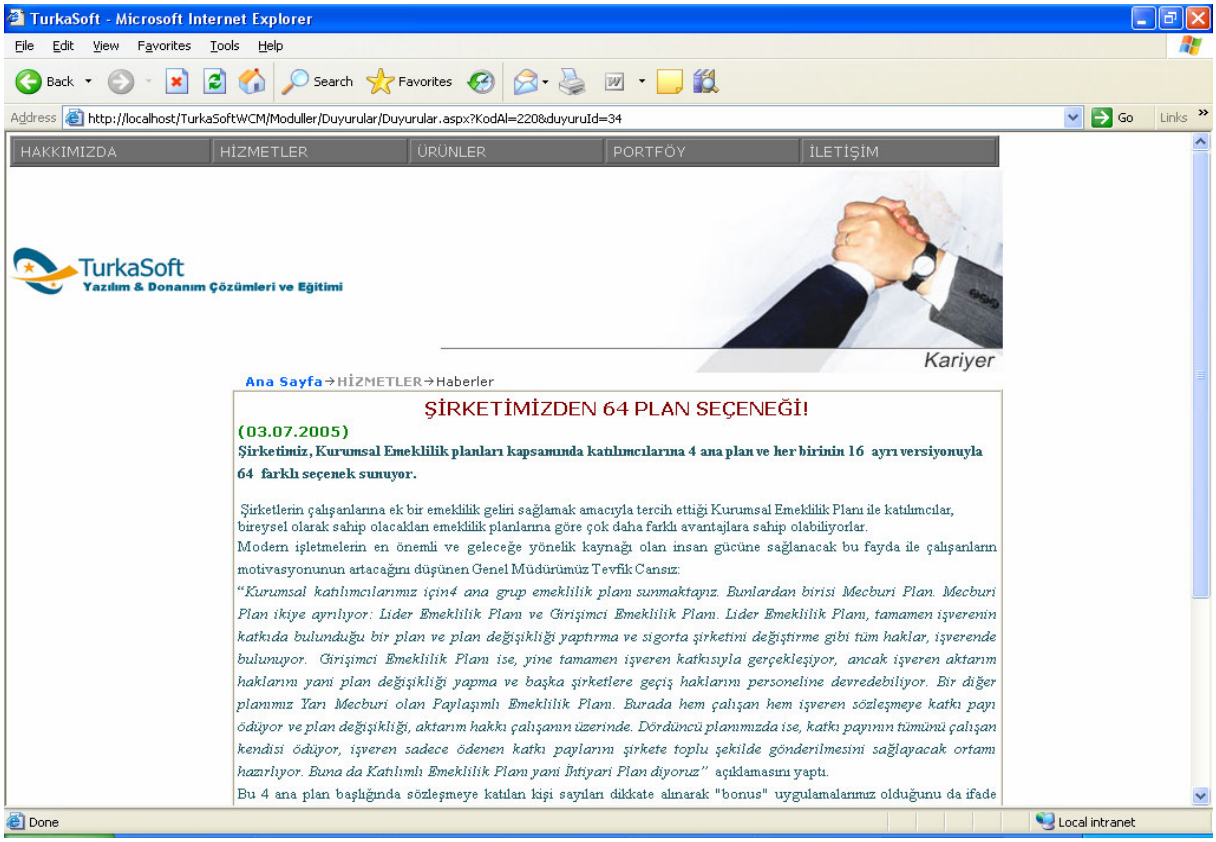


Figure 20 News Detail Page

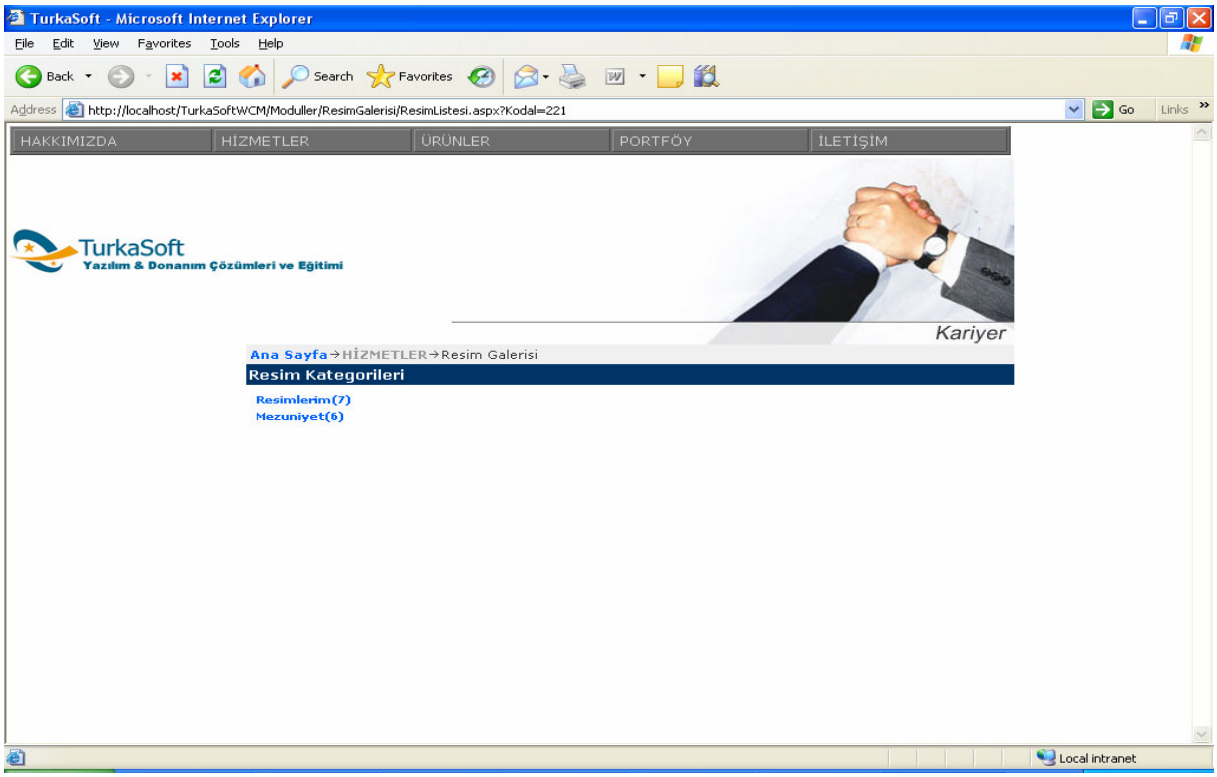


Figure 21 Picture Gallery Page

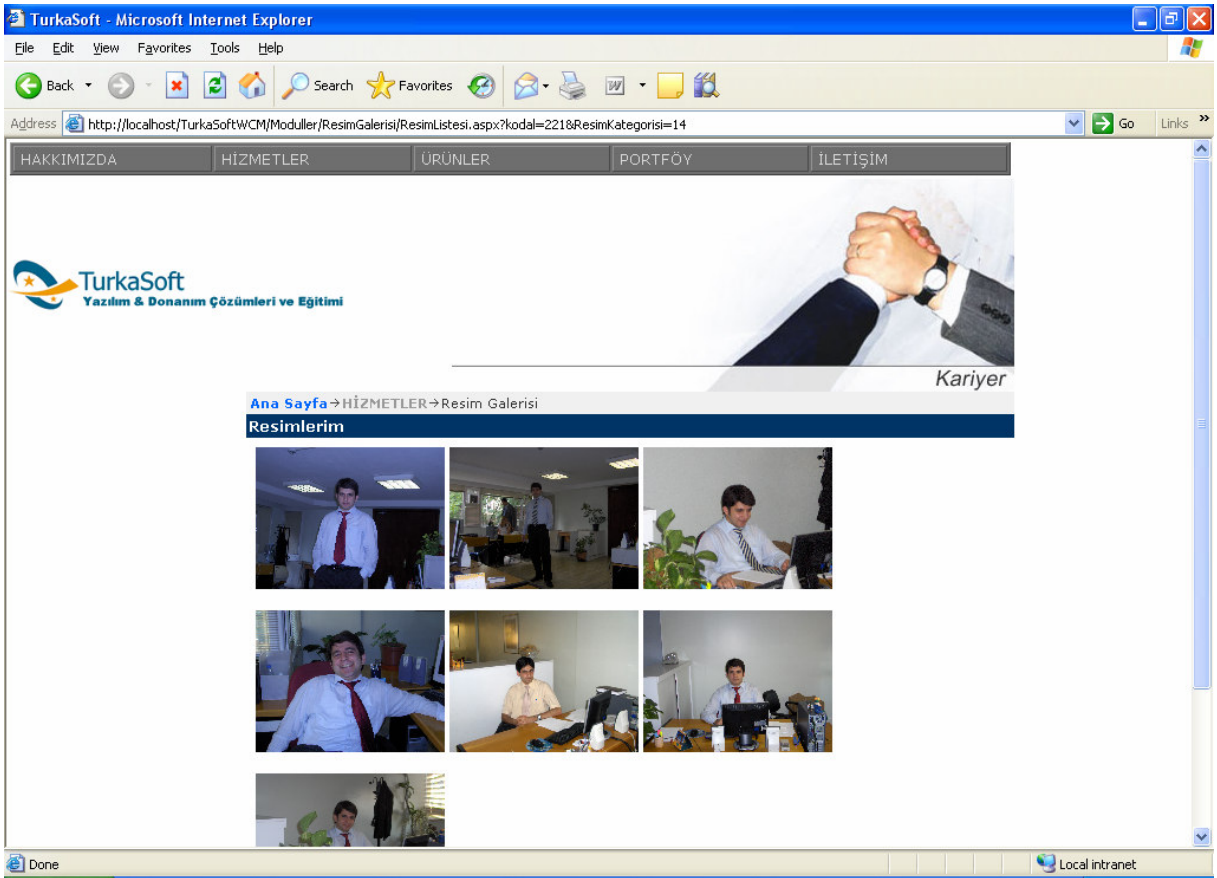


Figure 22 Picture Gallery Detail Page

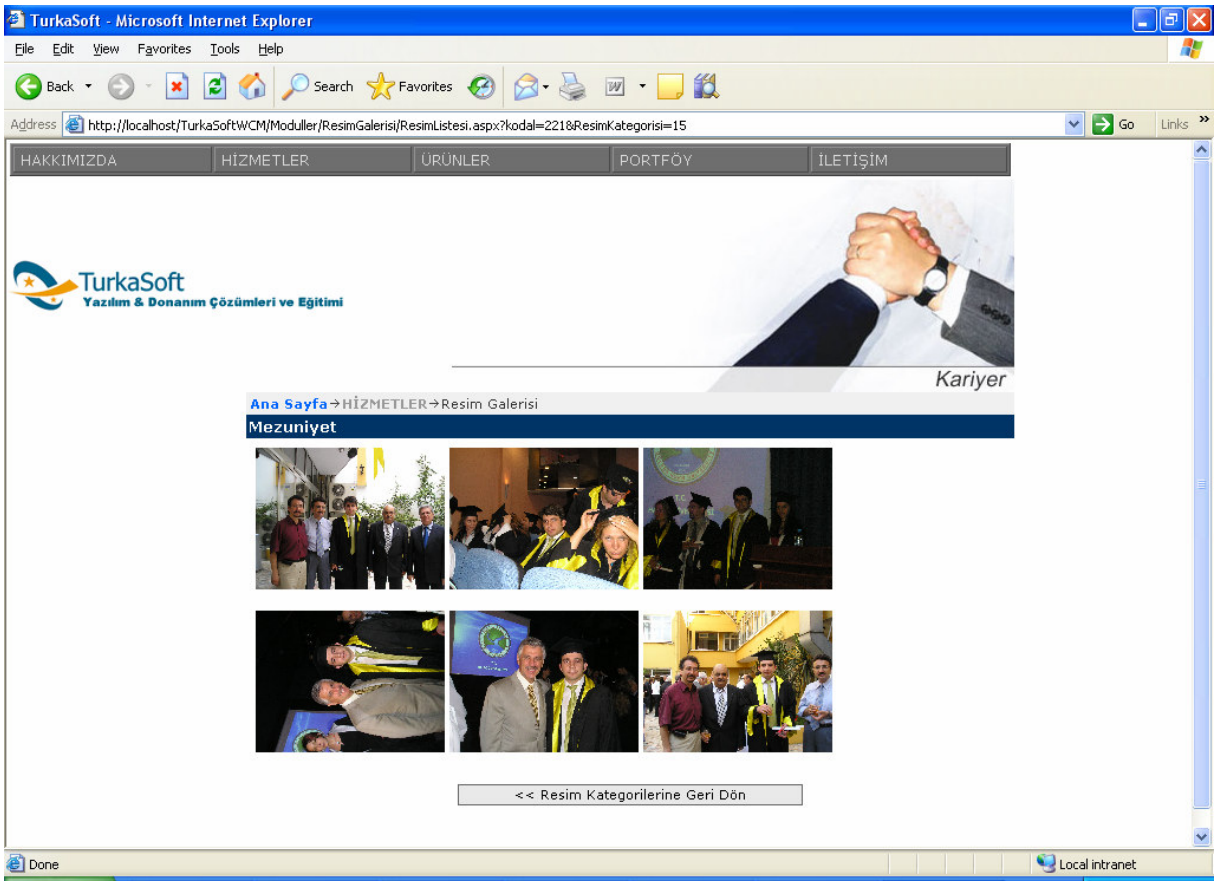


Figure 23 Picture Gallery Detail Page

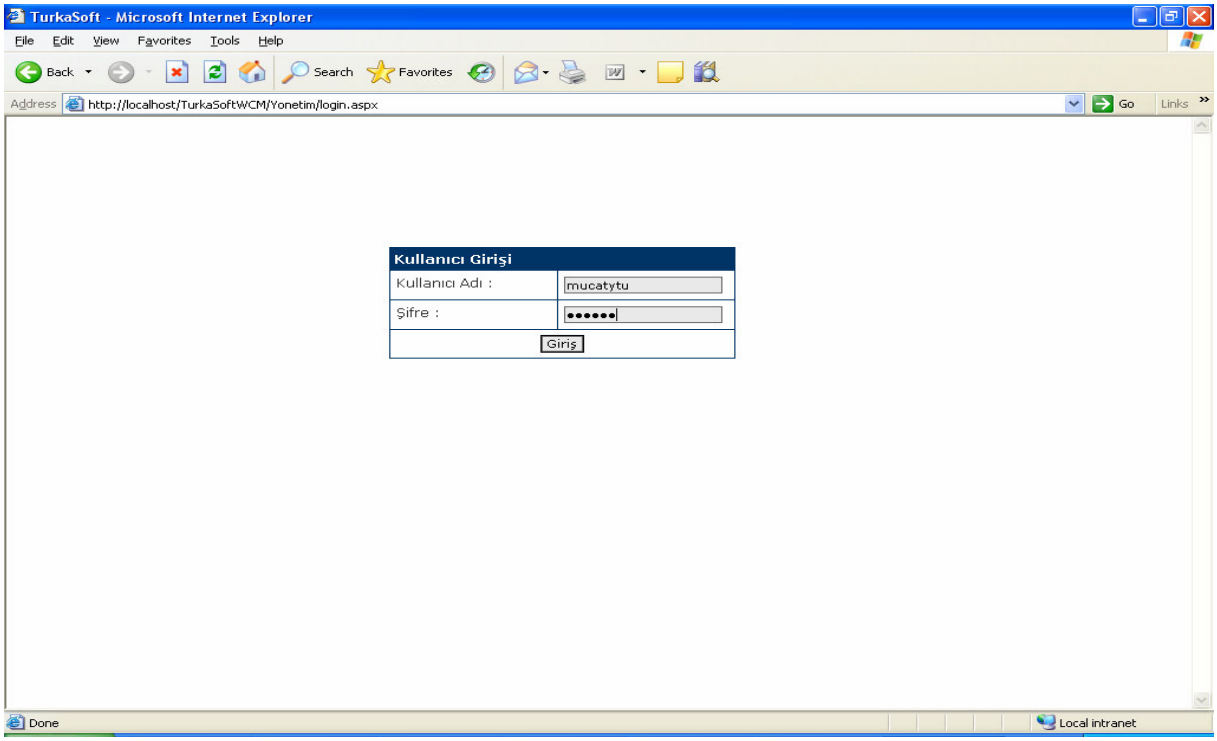


Figure 24 Admin Login Page

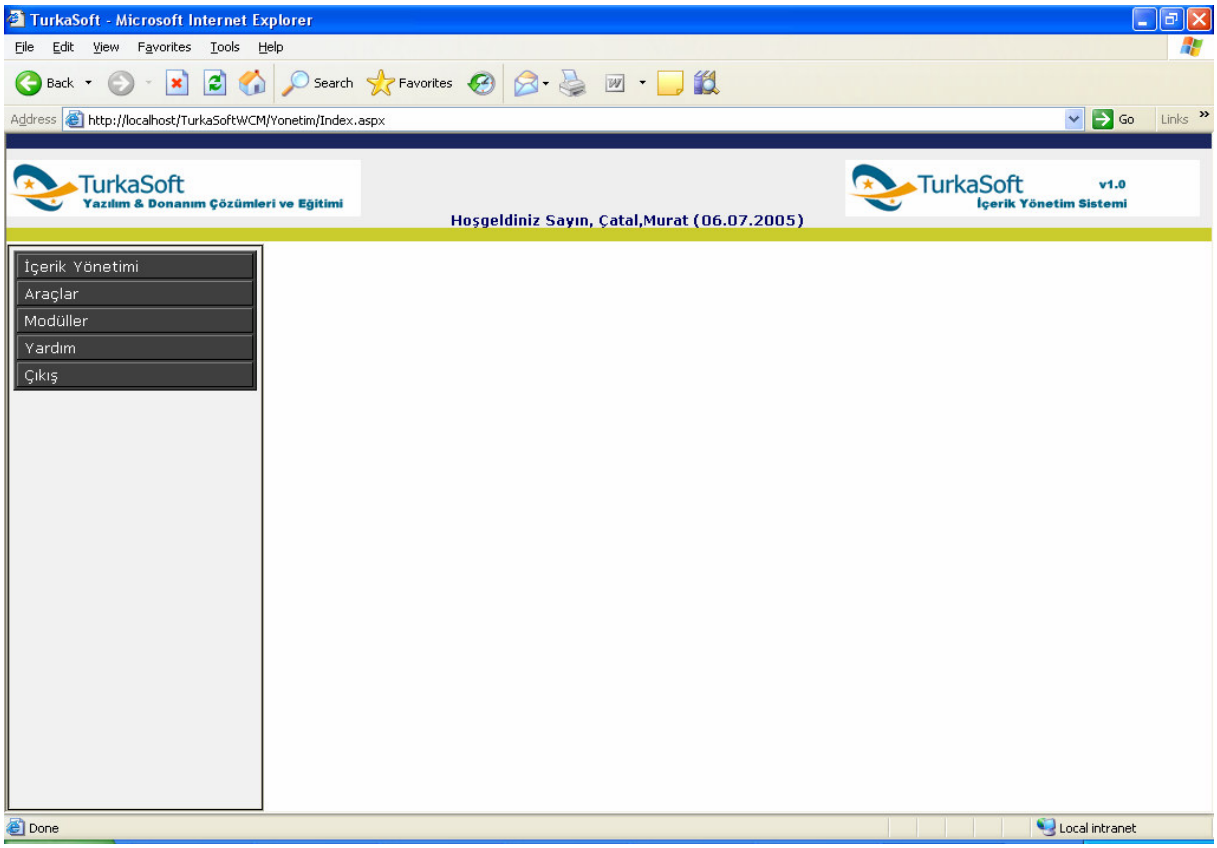


Figure 25 Admin Entrance Page

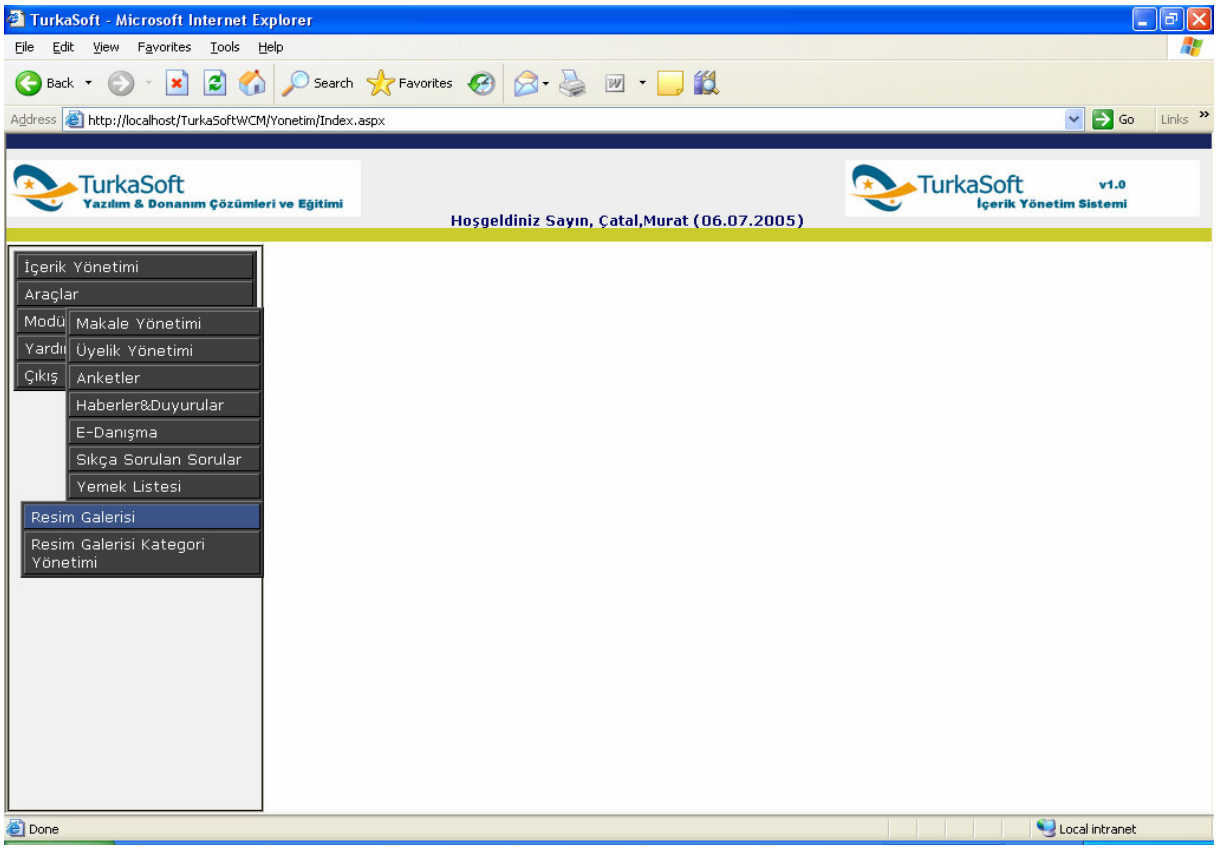


Figure 26 Admin Entrance Page

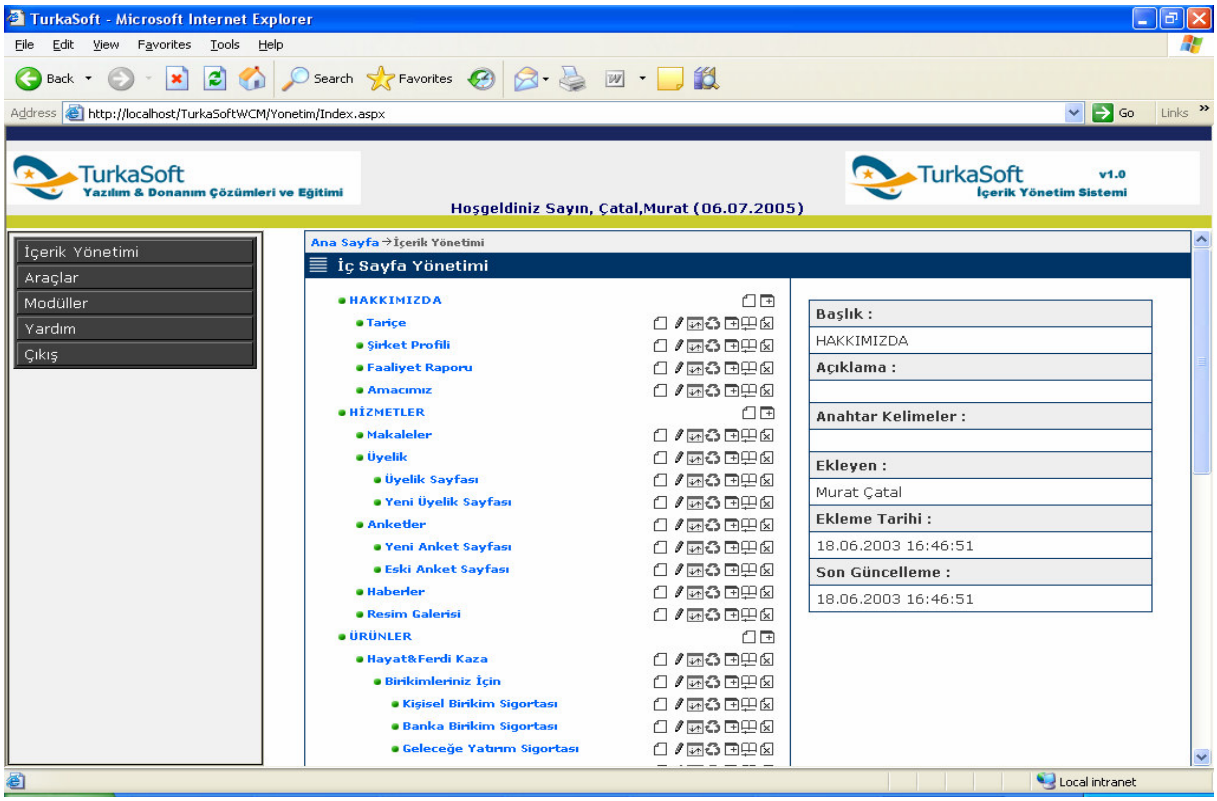


Figure 27 Interior Pages Management Page

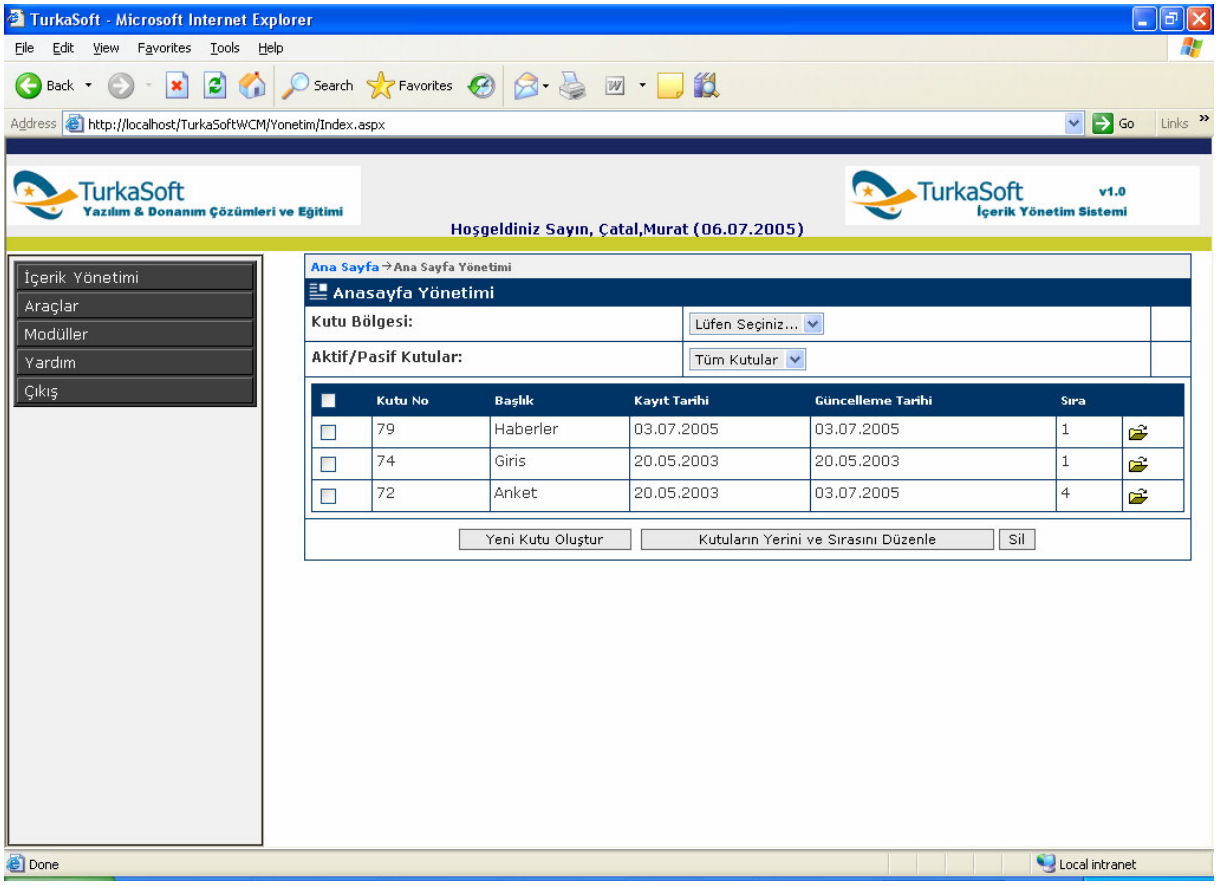


Figure 28 Homepage Management Page

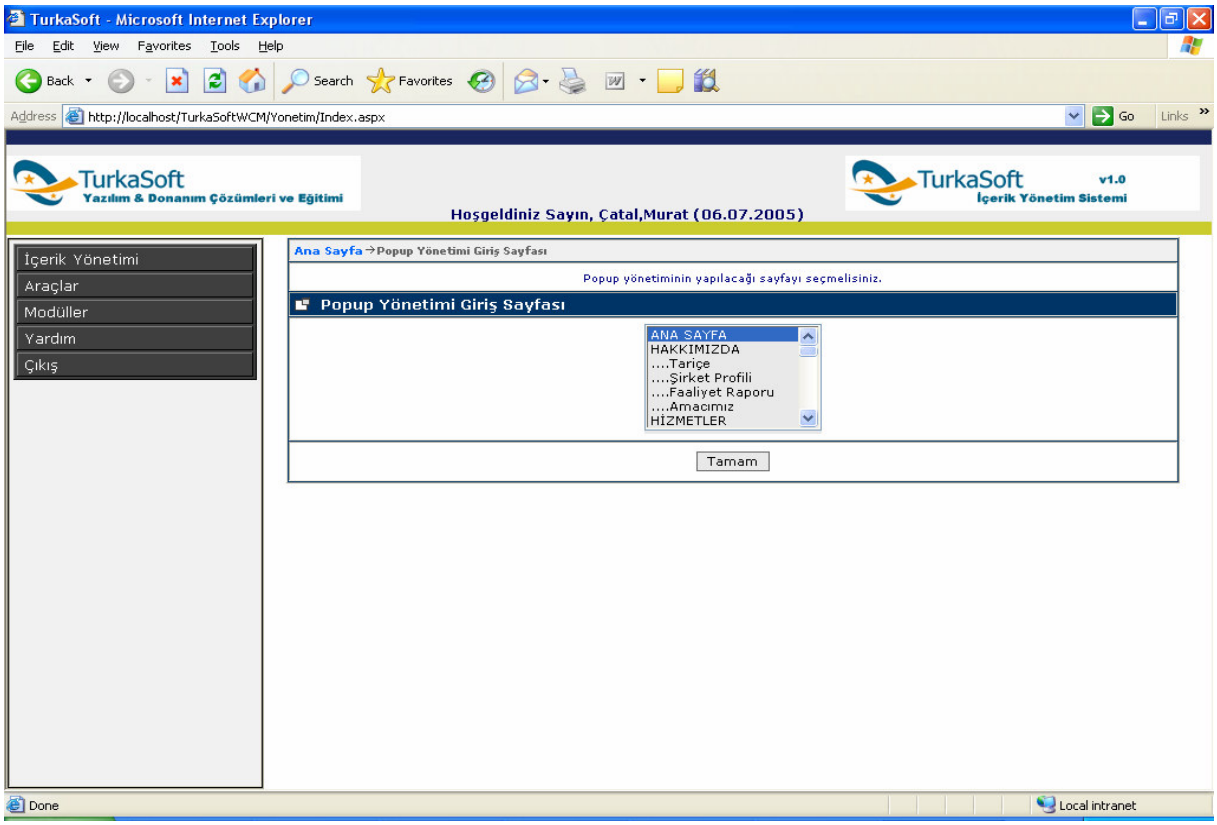


Figure 29 Popup Management Page

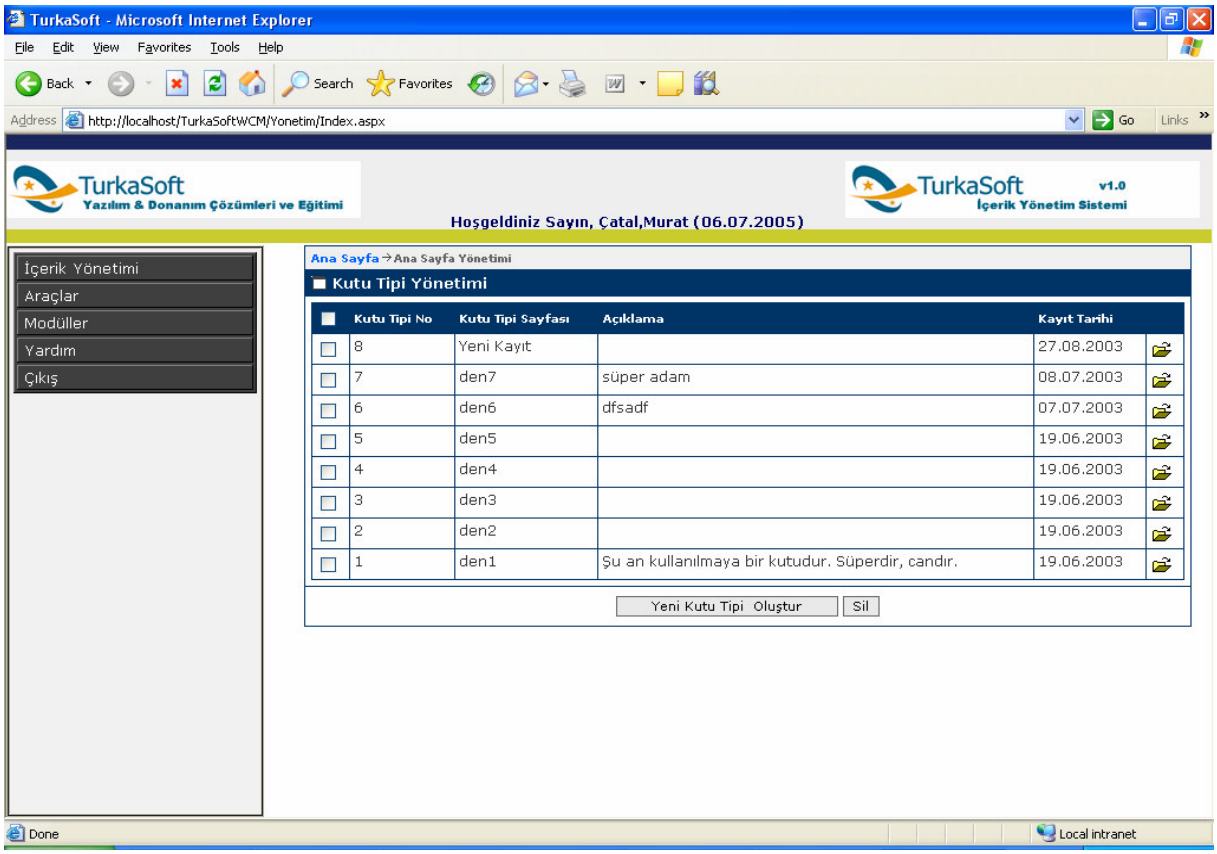


Figure 30 Box Managament Page

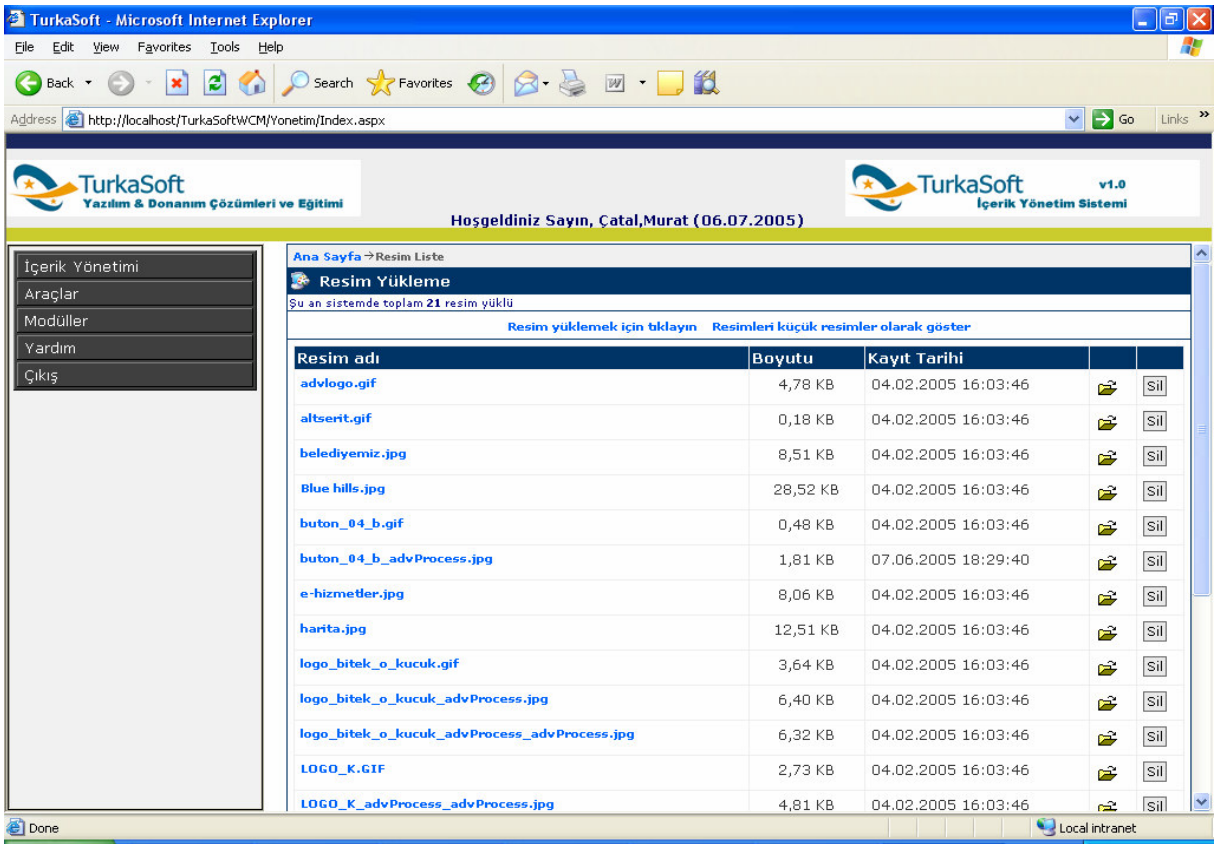


Figure 31 Picture Management Page

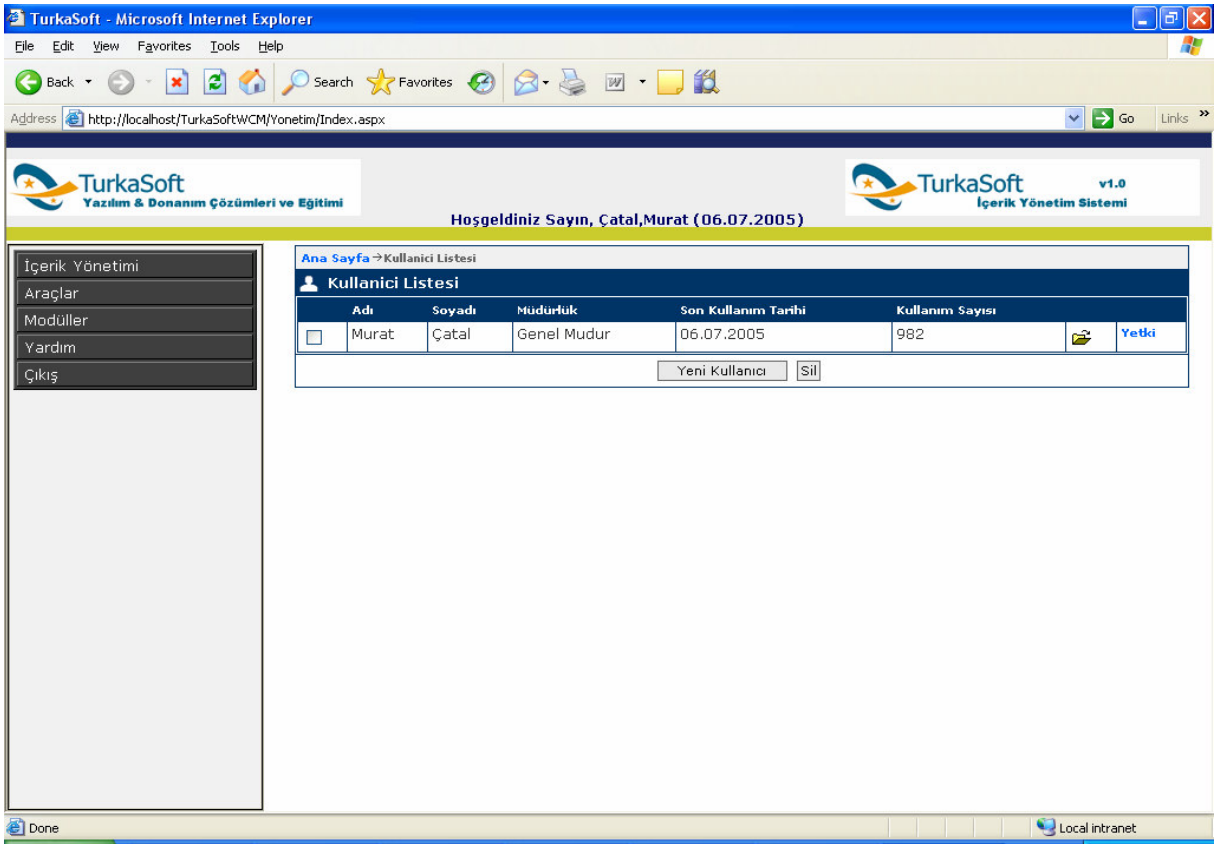


Figure 32 User Management Page

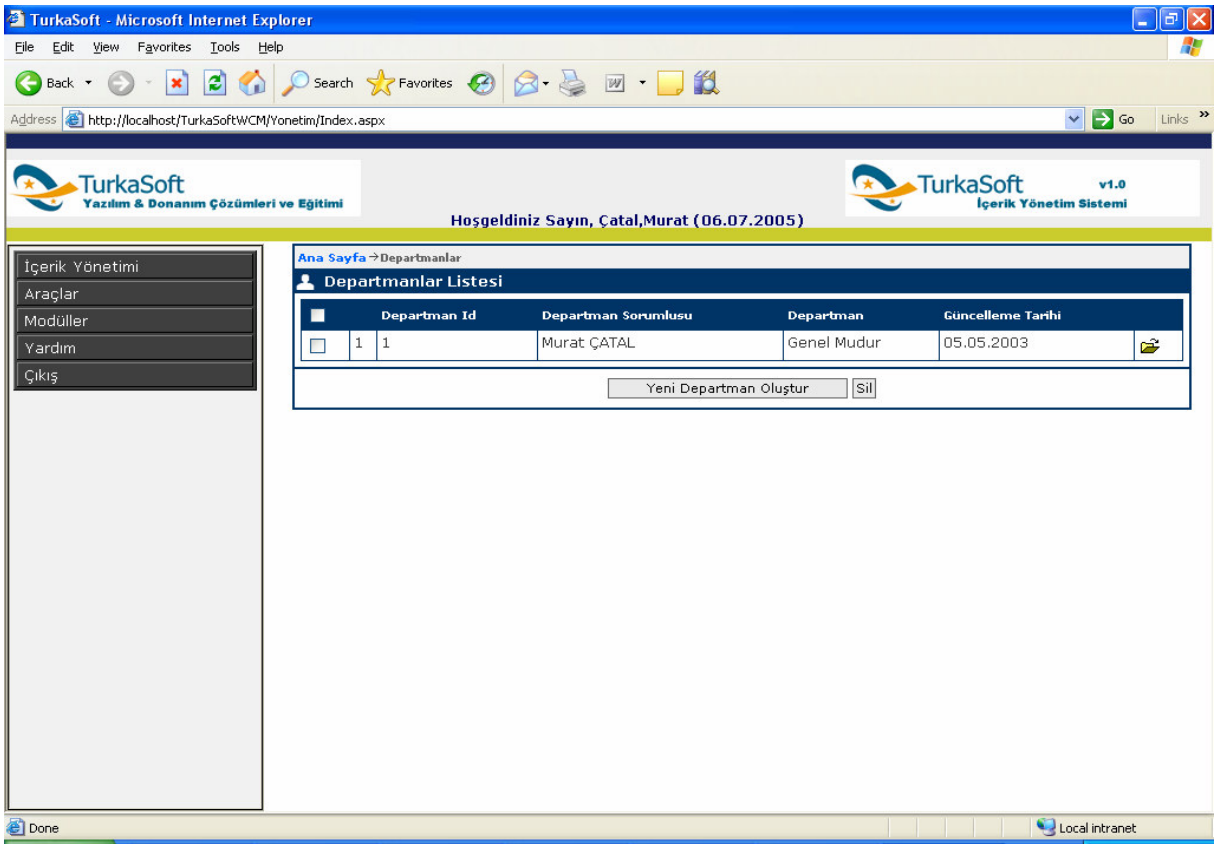


Figure 33 Department Management Page

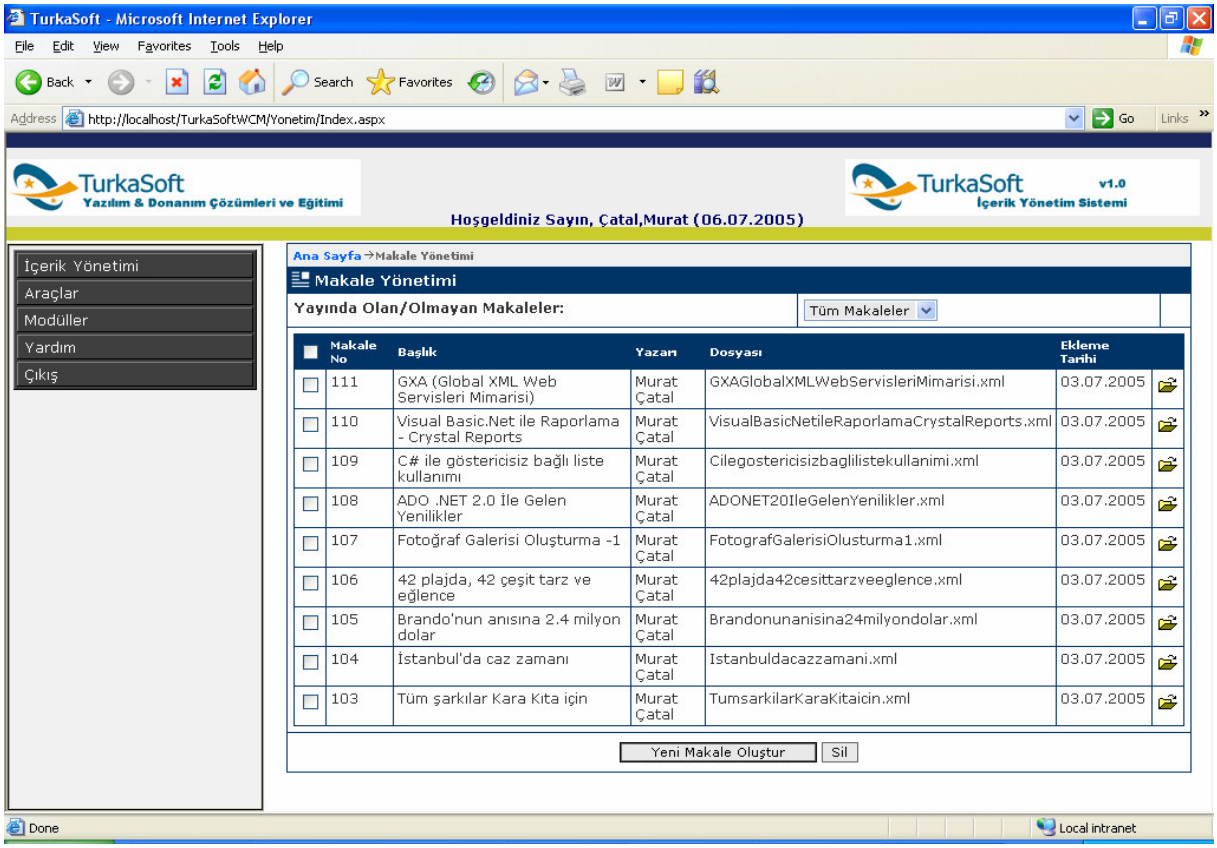


Figure 34 Article Content Management Page

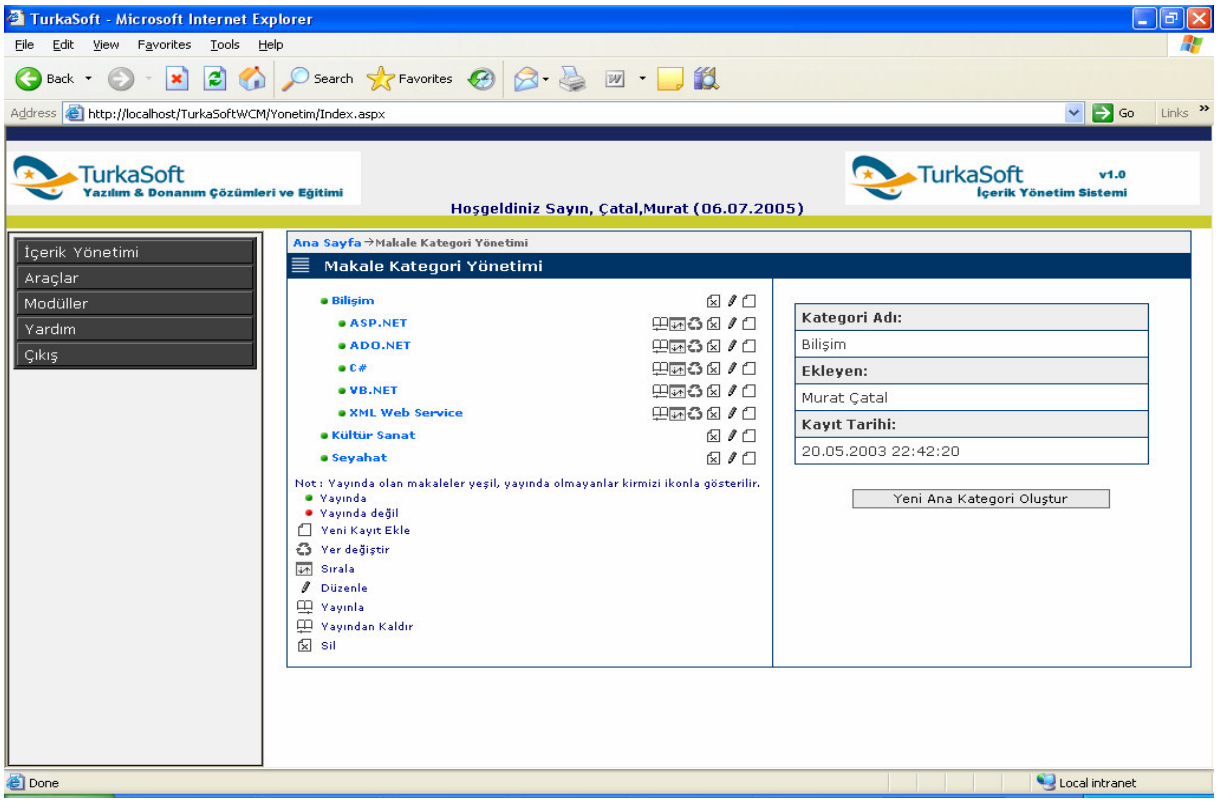


Figure 35 Article Category Management Page

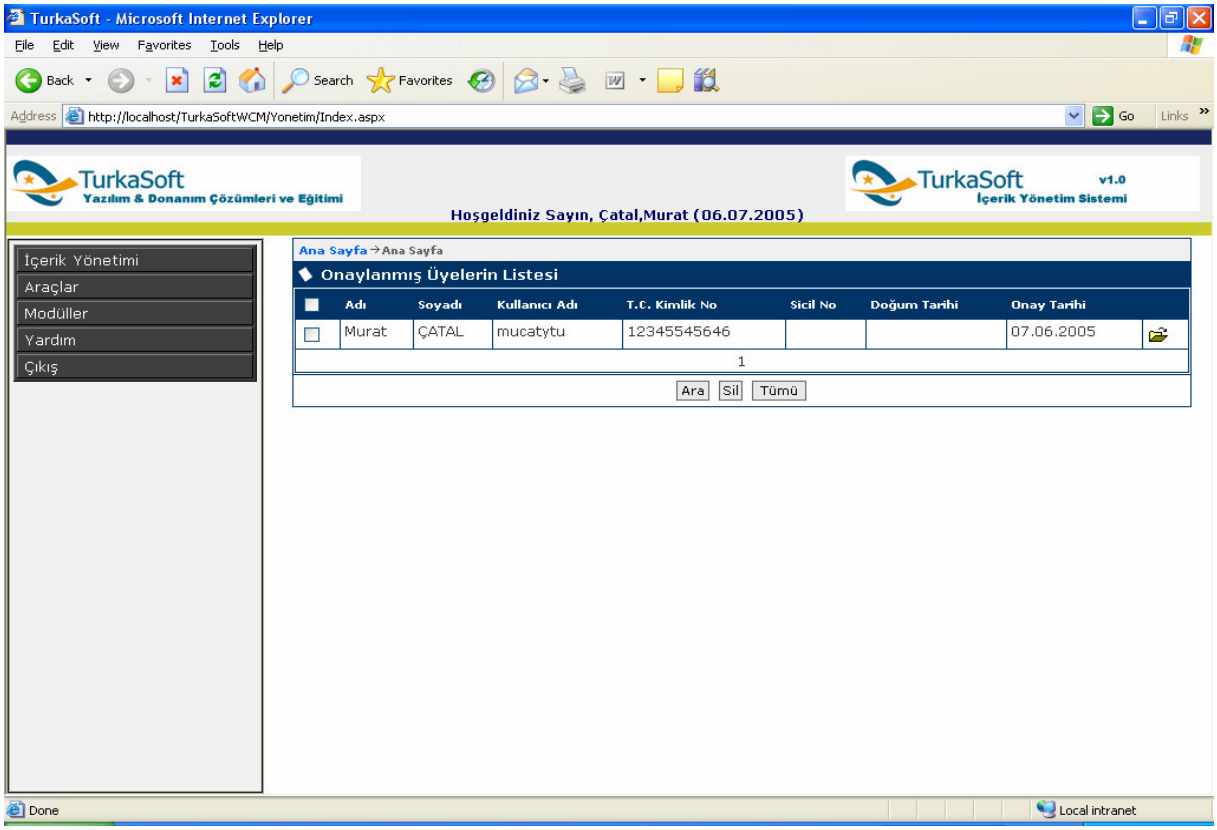


Figure 36 Membership Management Page

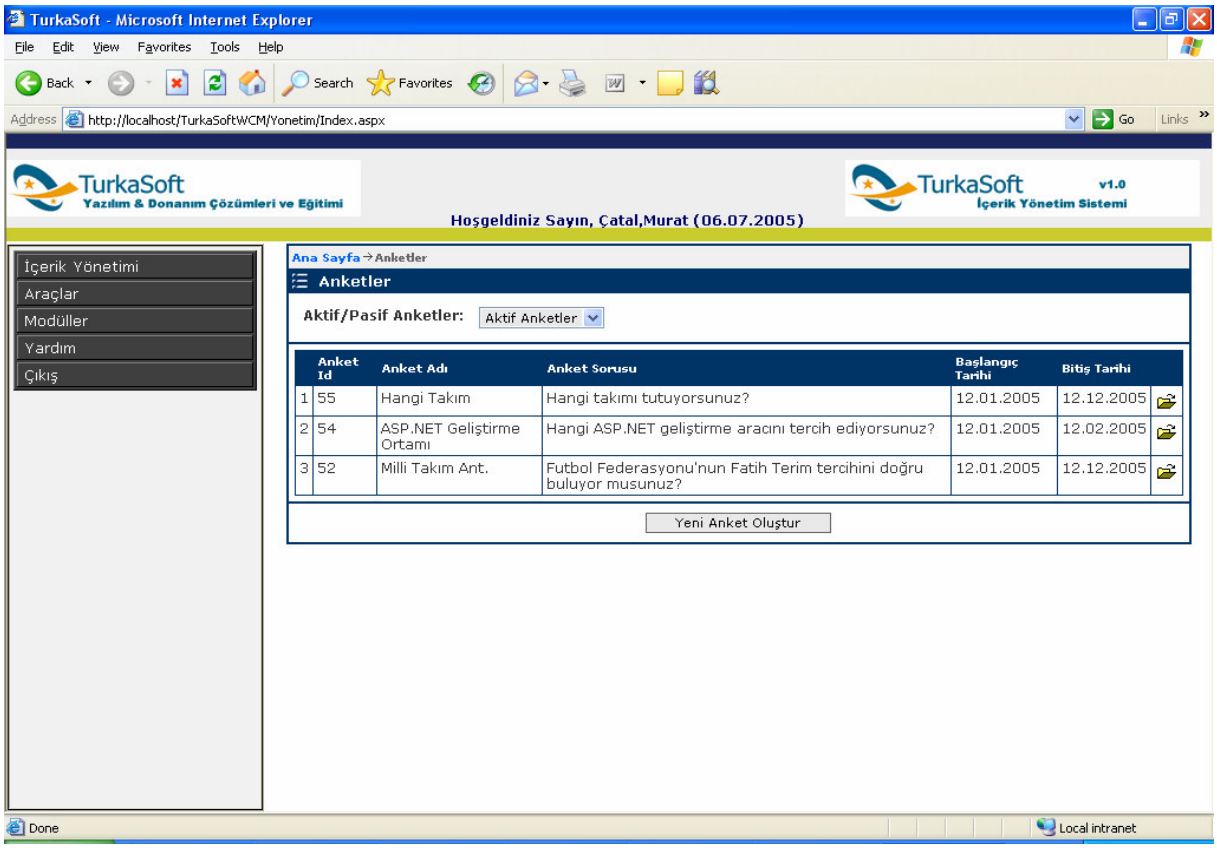


Figure 37 Poll Management Page

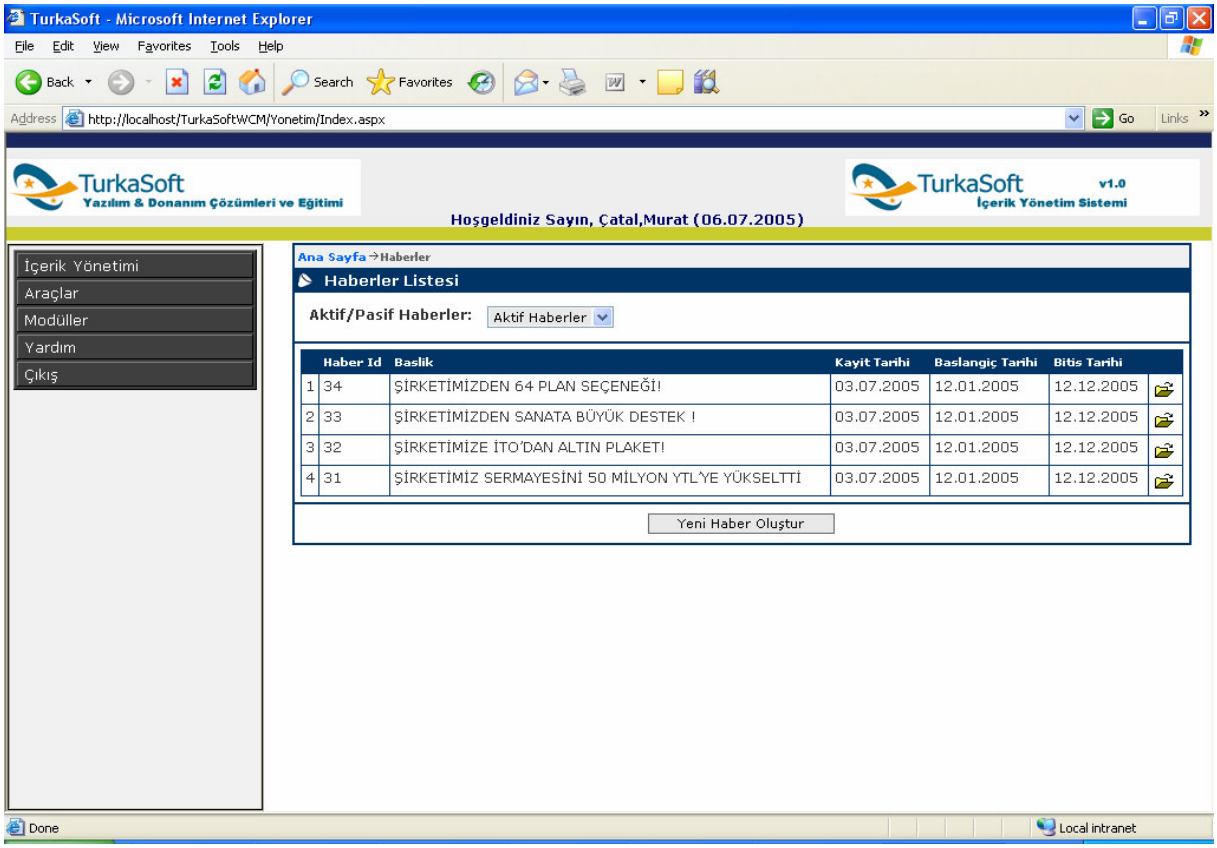


Figure 38 News Management Page

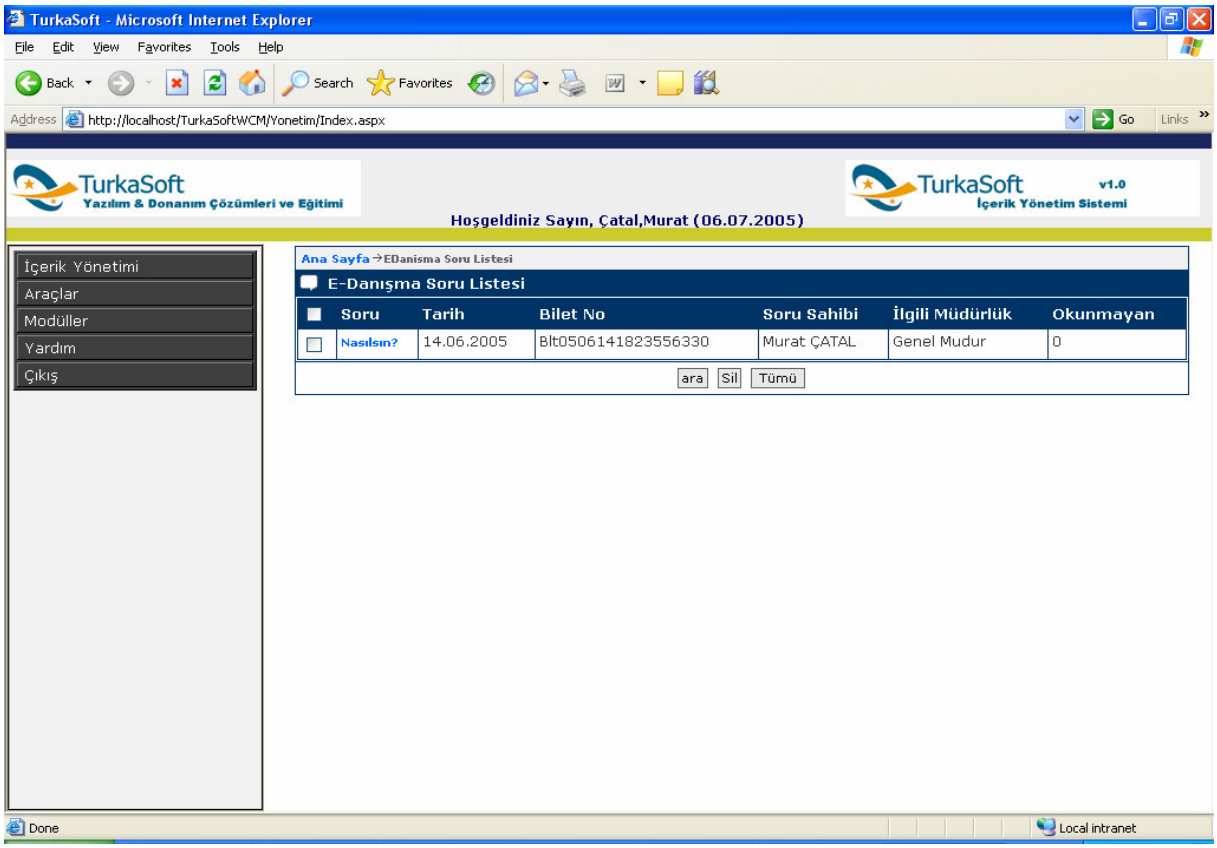


Figure 39 E-Support Management Page

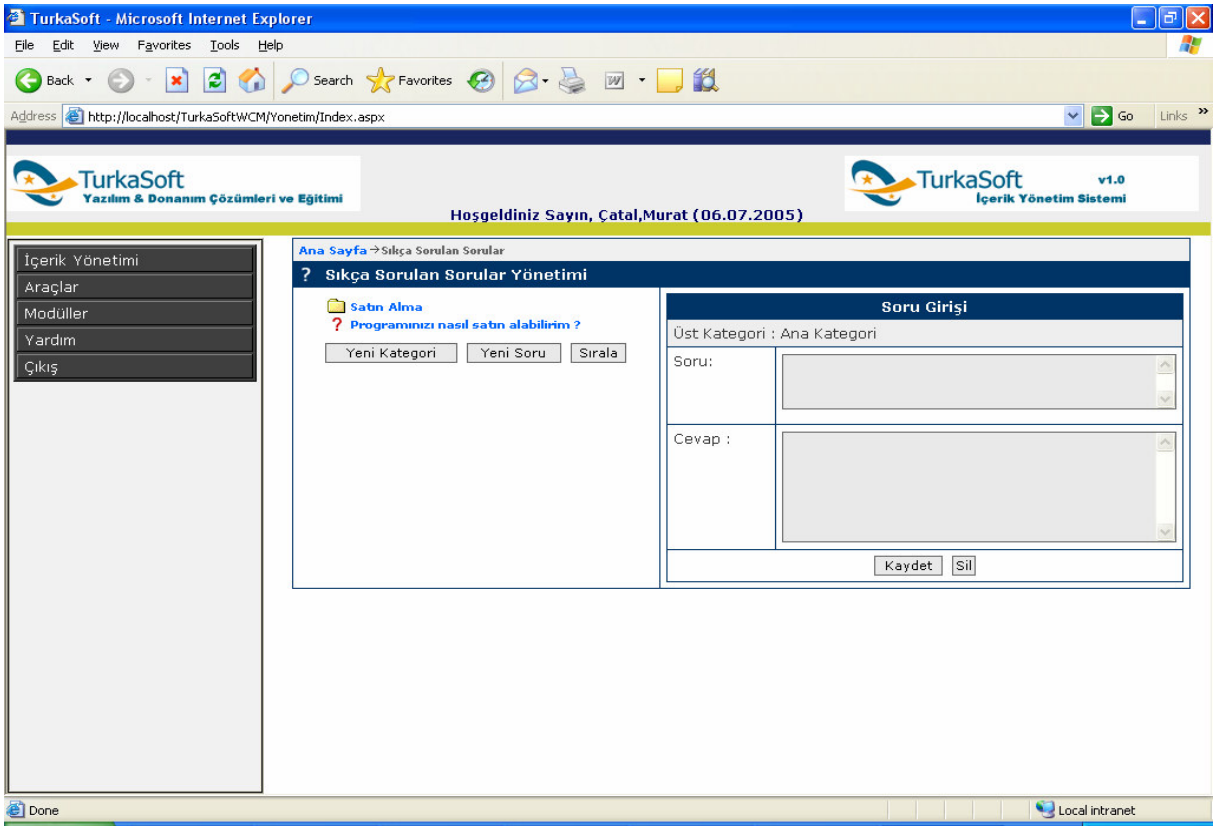


Figure 40 FAQ Management Page

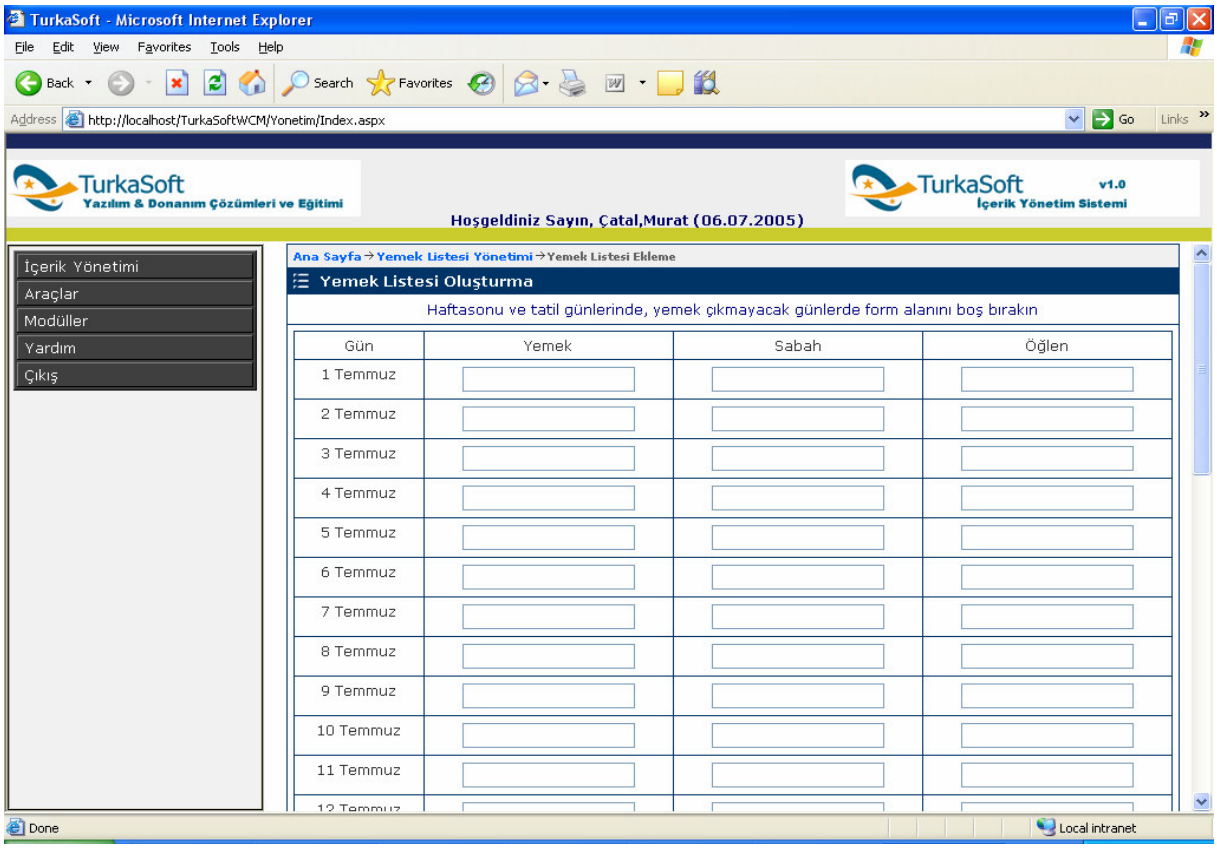


Figure 41 Food Management Page

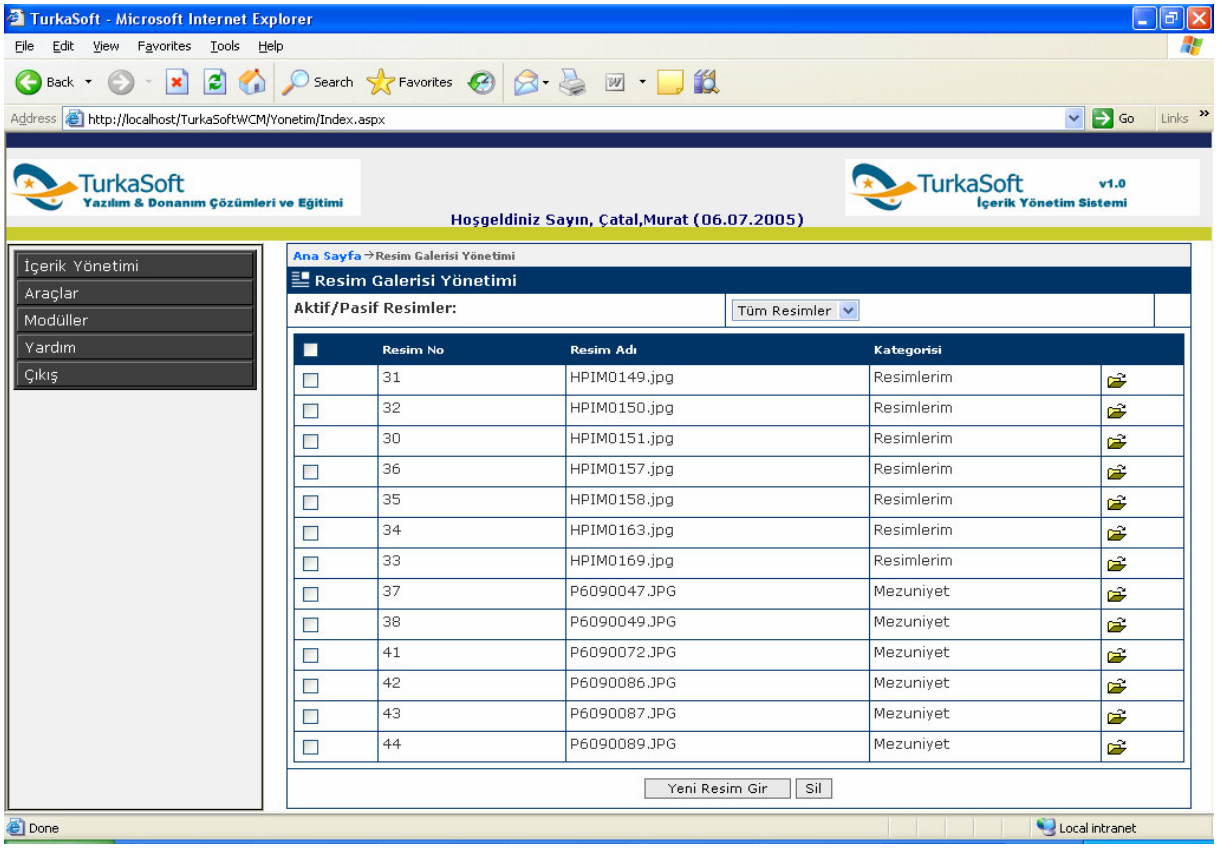


Figure 42 Picture Gallery Management Page

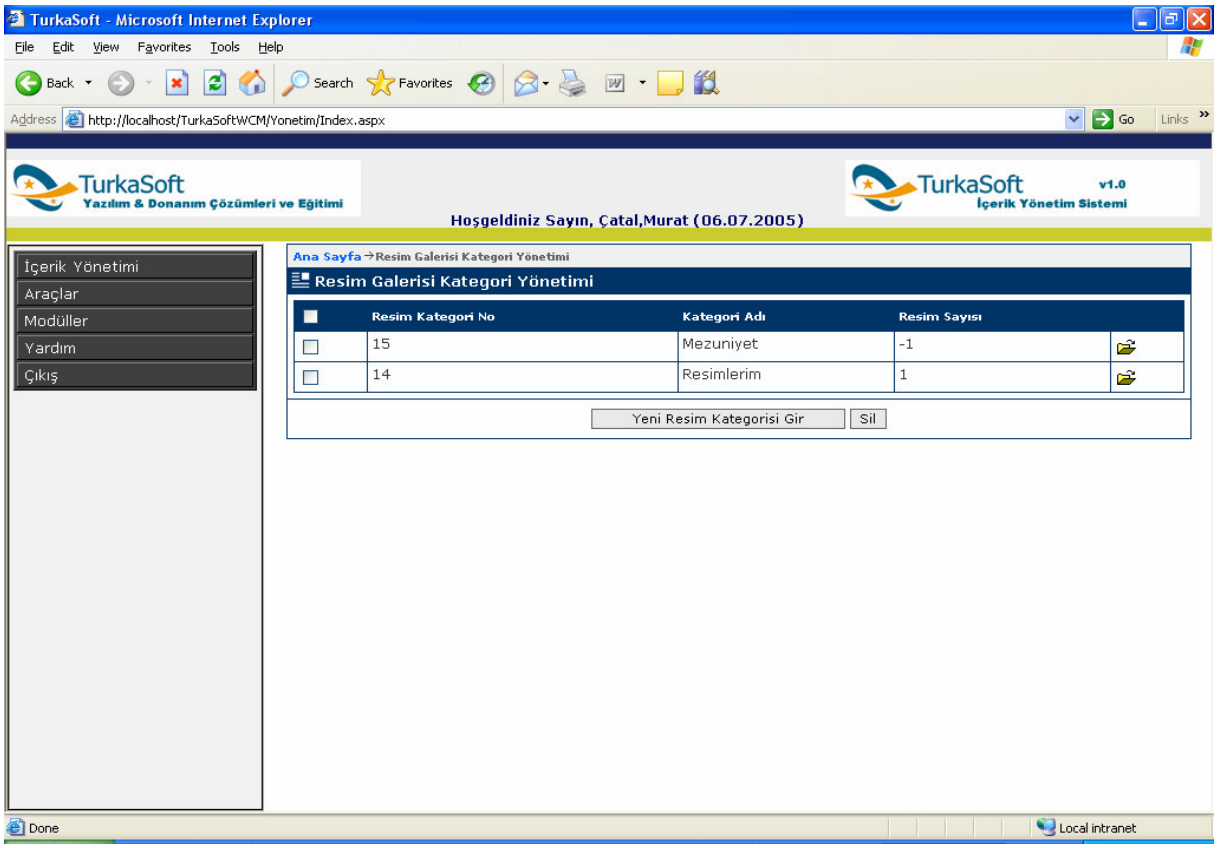


Figure 43 Picture Gallery Category Management Page

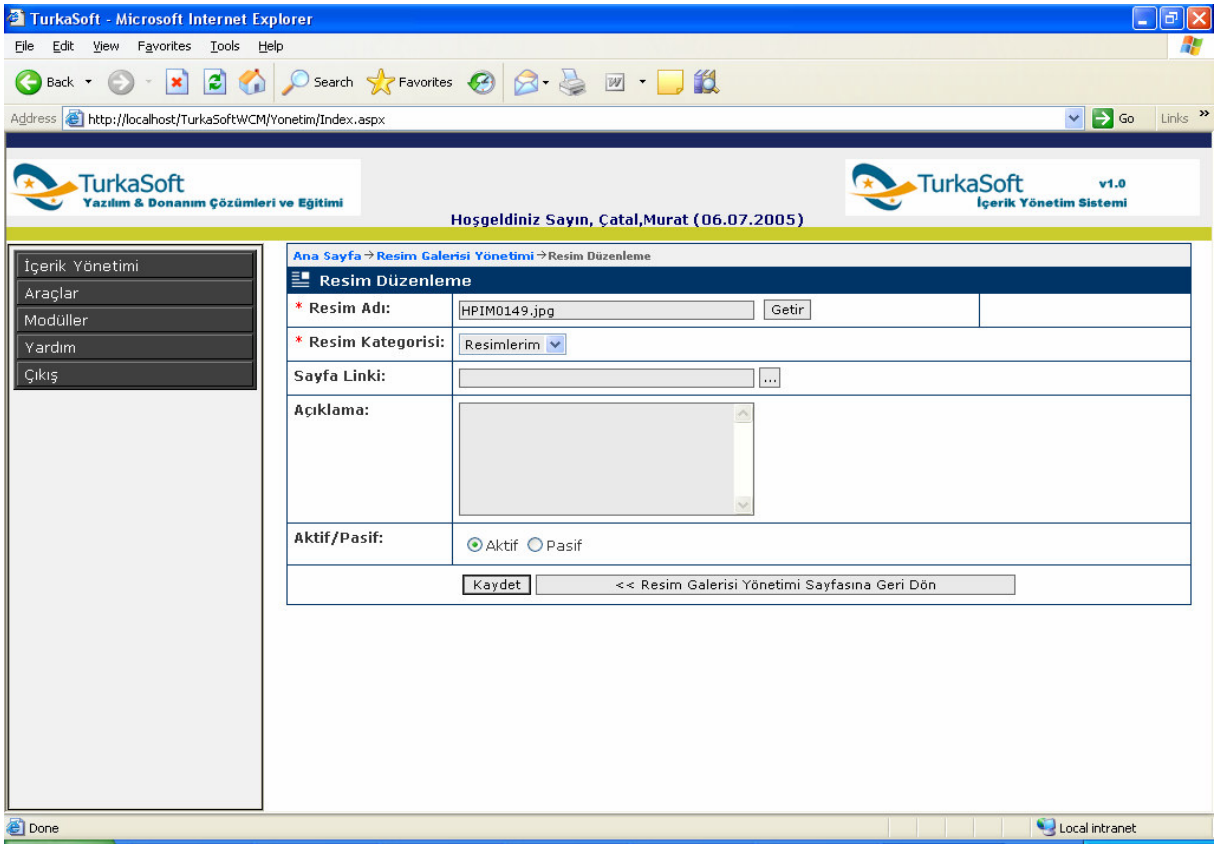


Figure 44 Picture Update Page

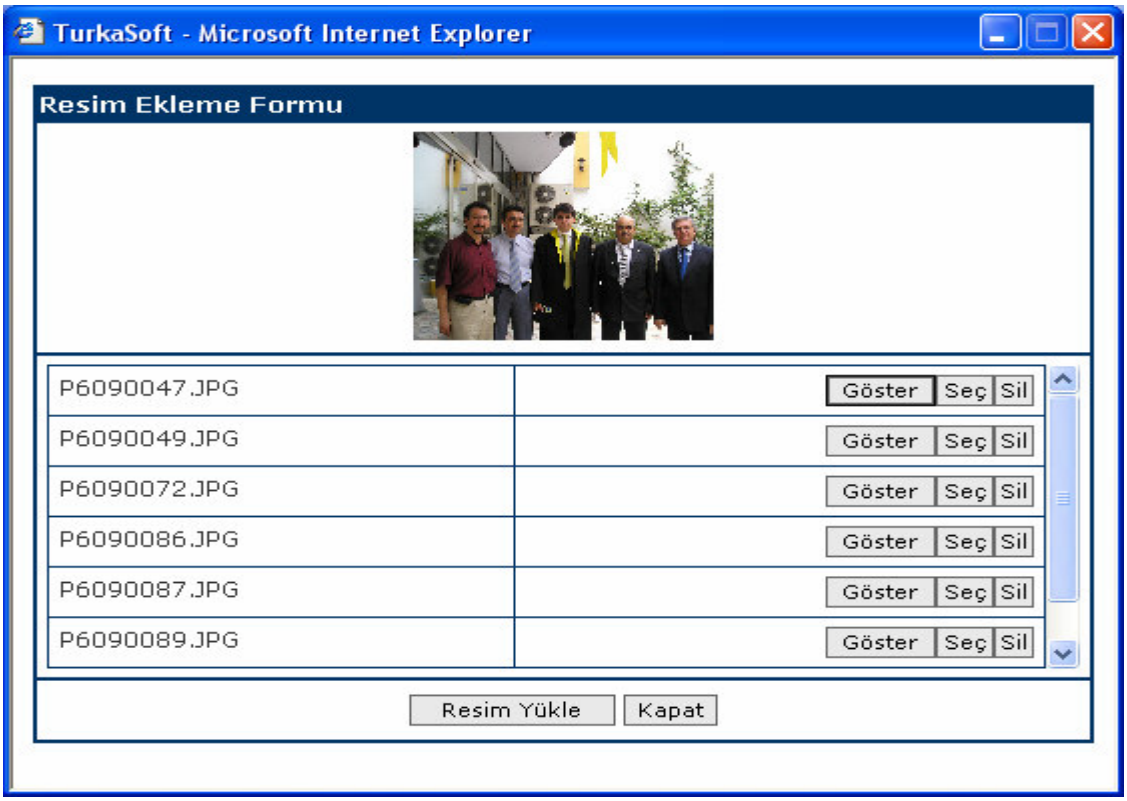


Figure 45 Upload Picture Page

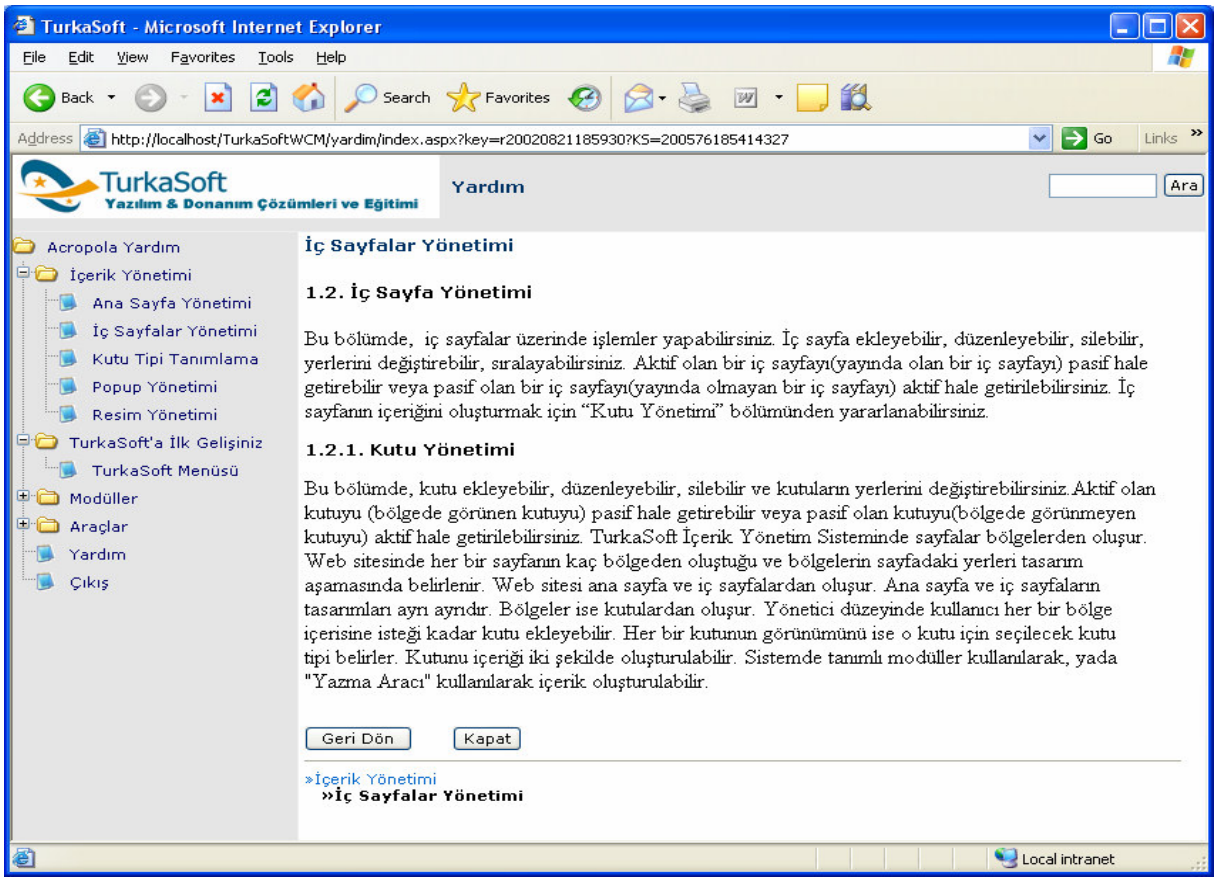


Figure 46 COMAS(Content Management System) Help Pages

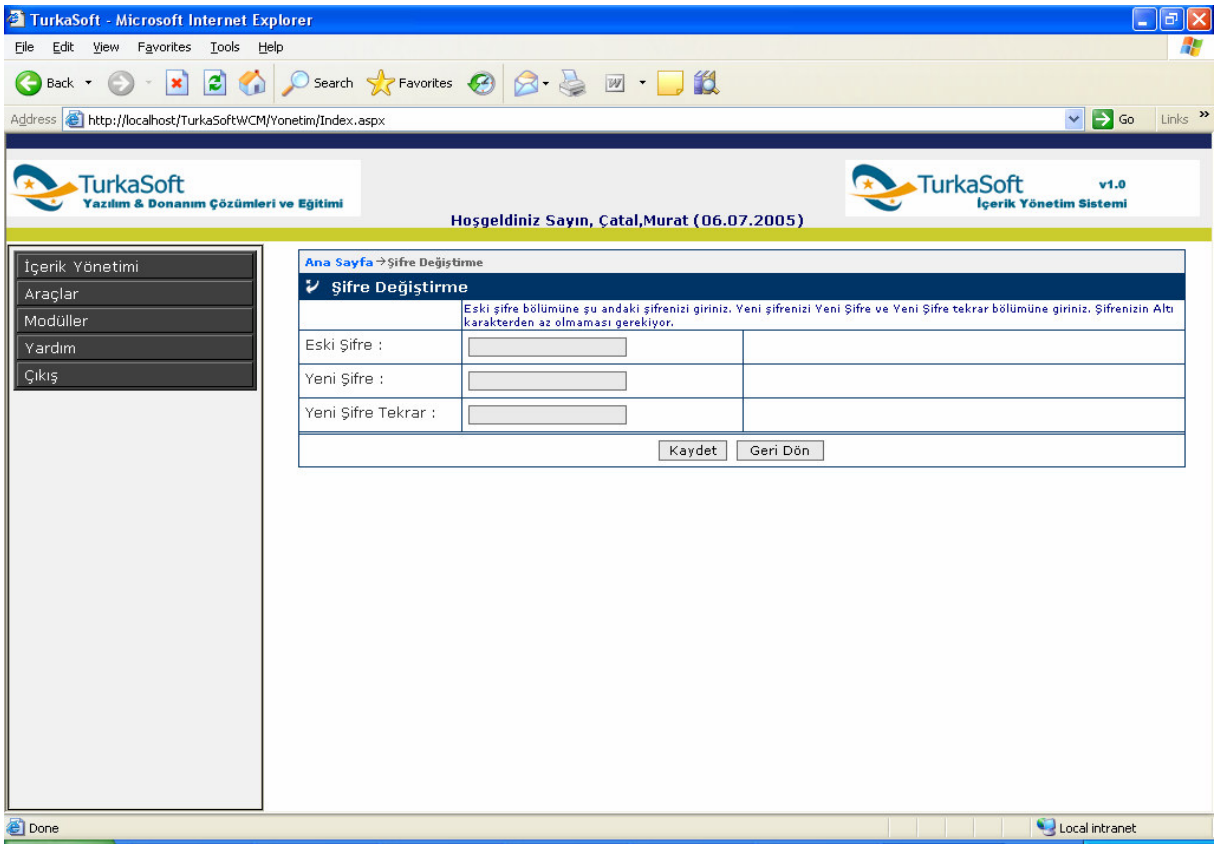


Figure 47 User Password Change Page

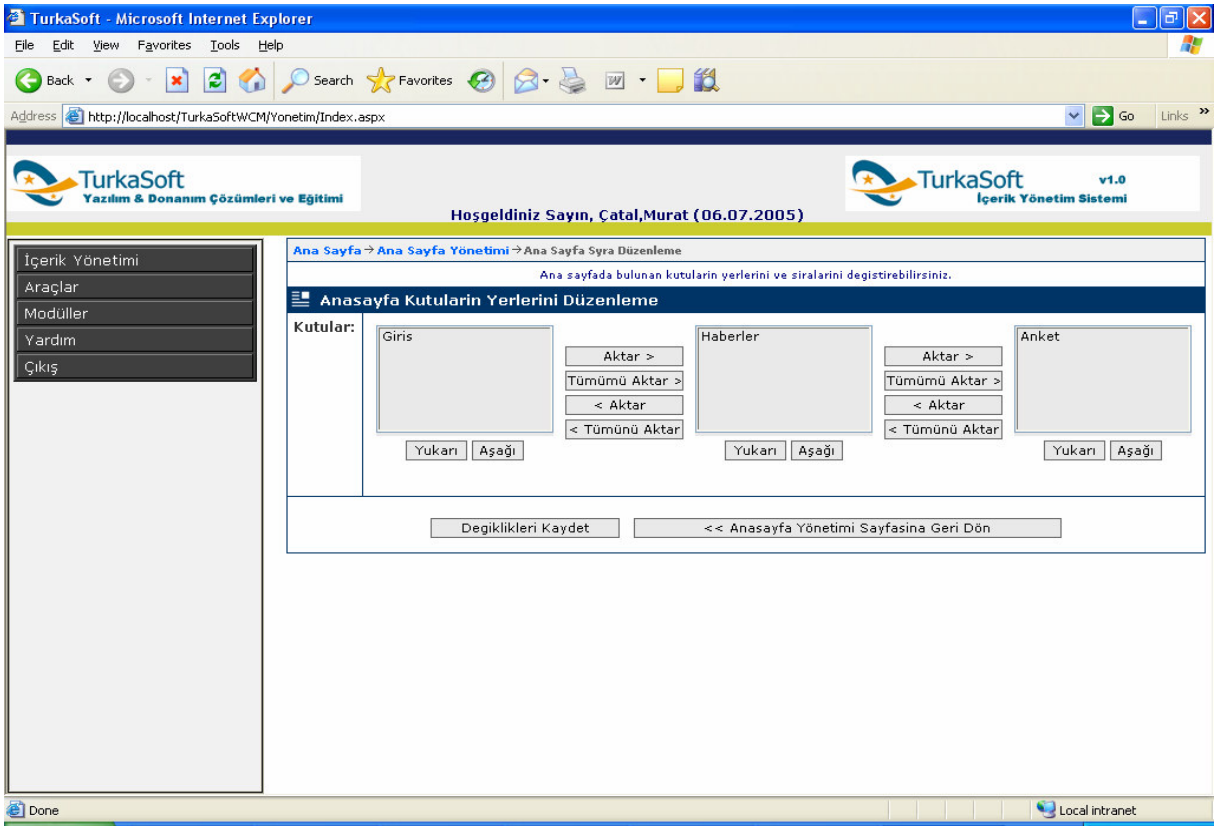


Figure 48 Homepage Management Page

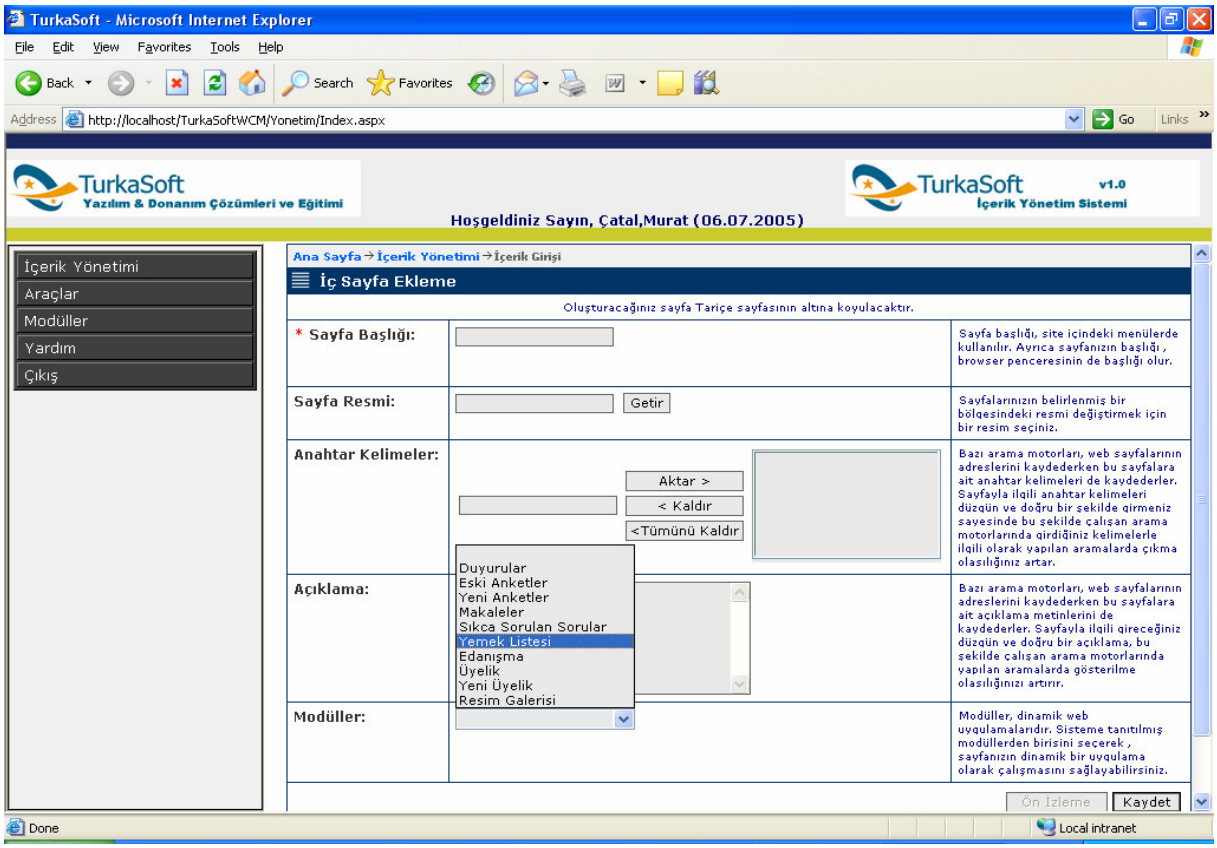


Figure 49 Interior Pages Management Page

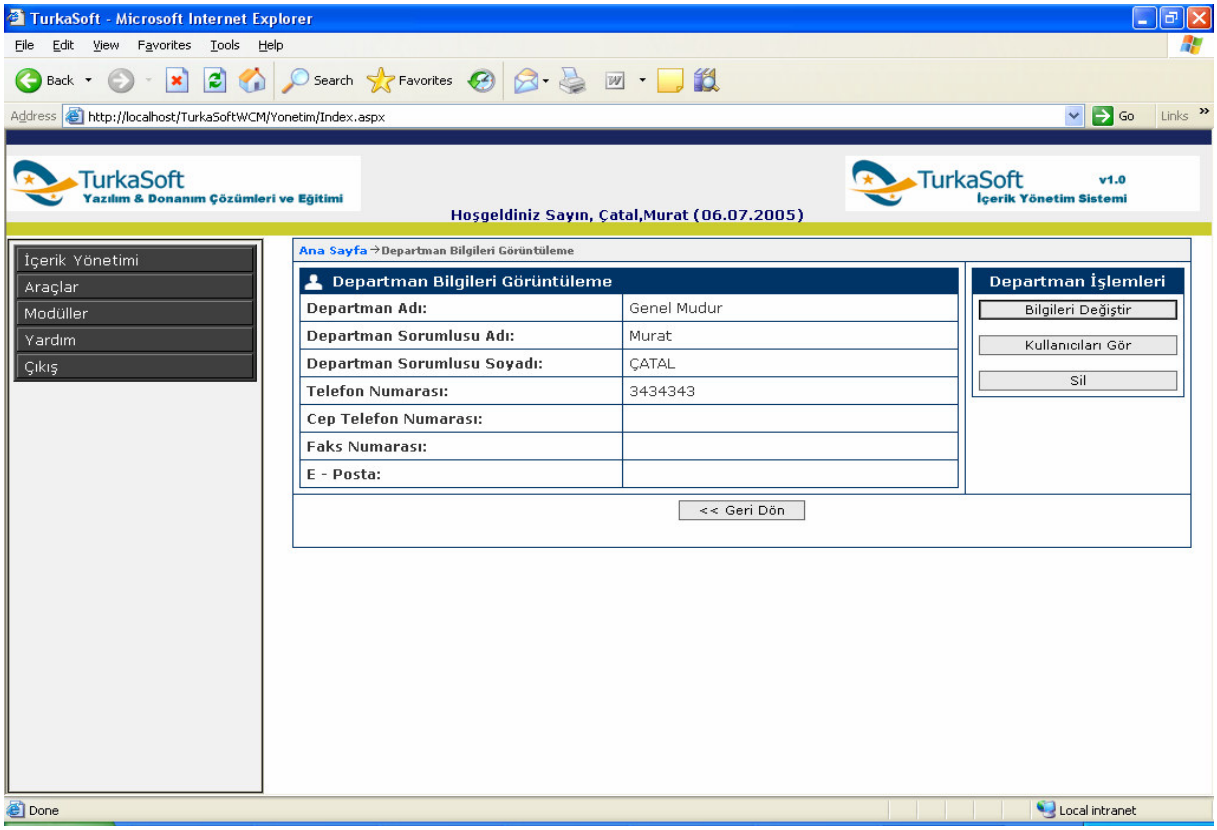


Figure 50 Department Update Page

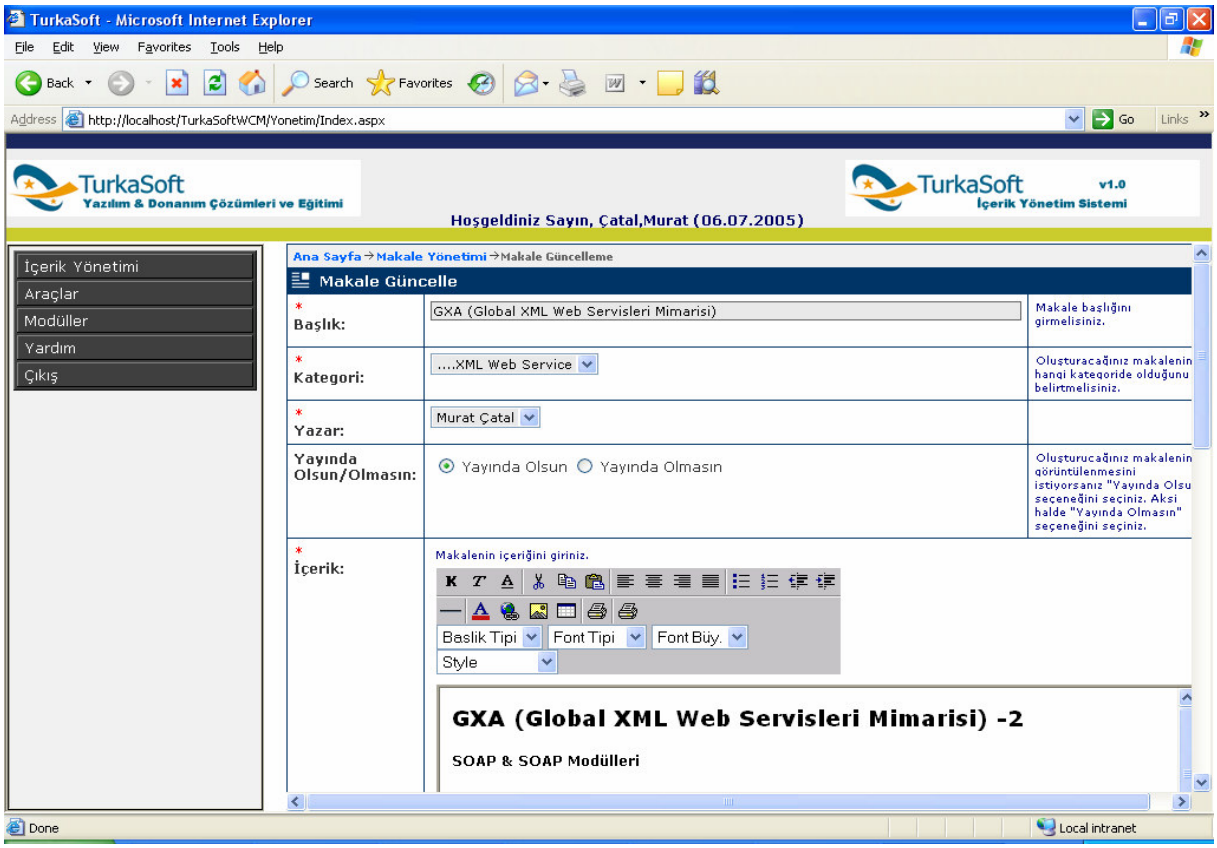


Figure 51 Article Update Page

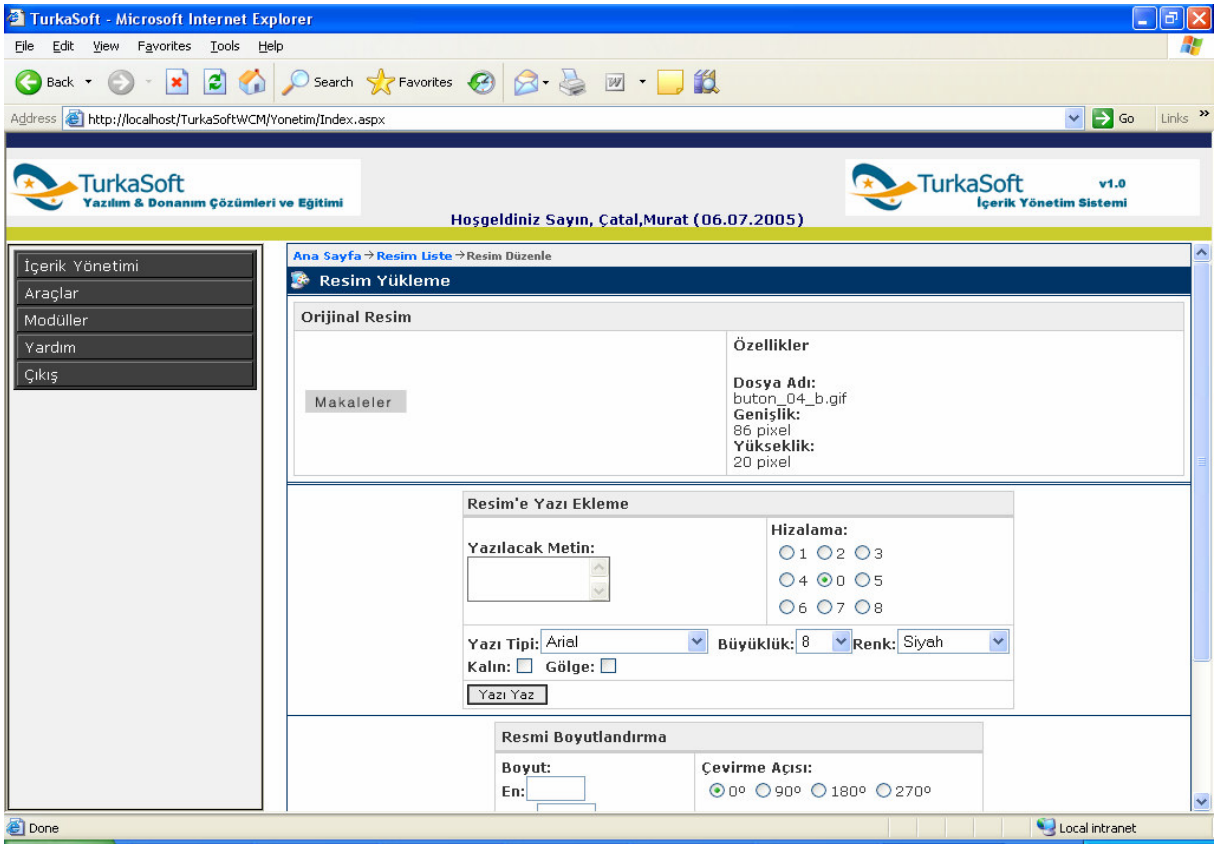


Figure 52 Picture Update Page

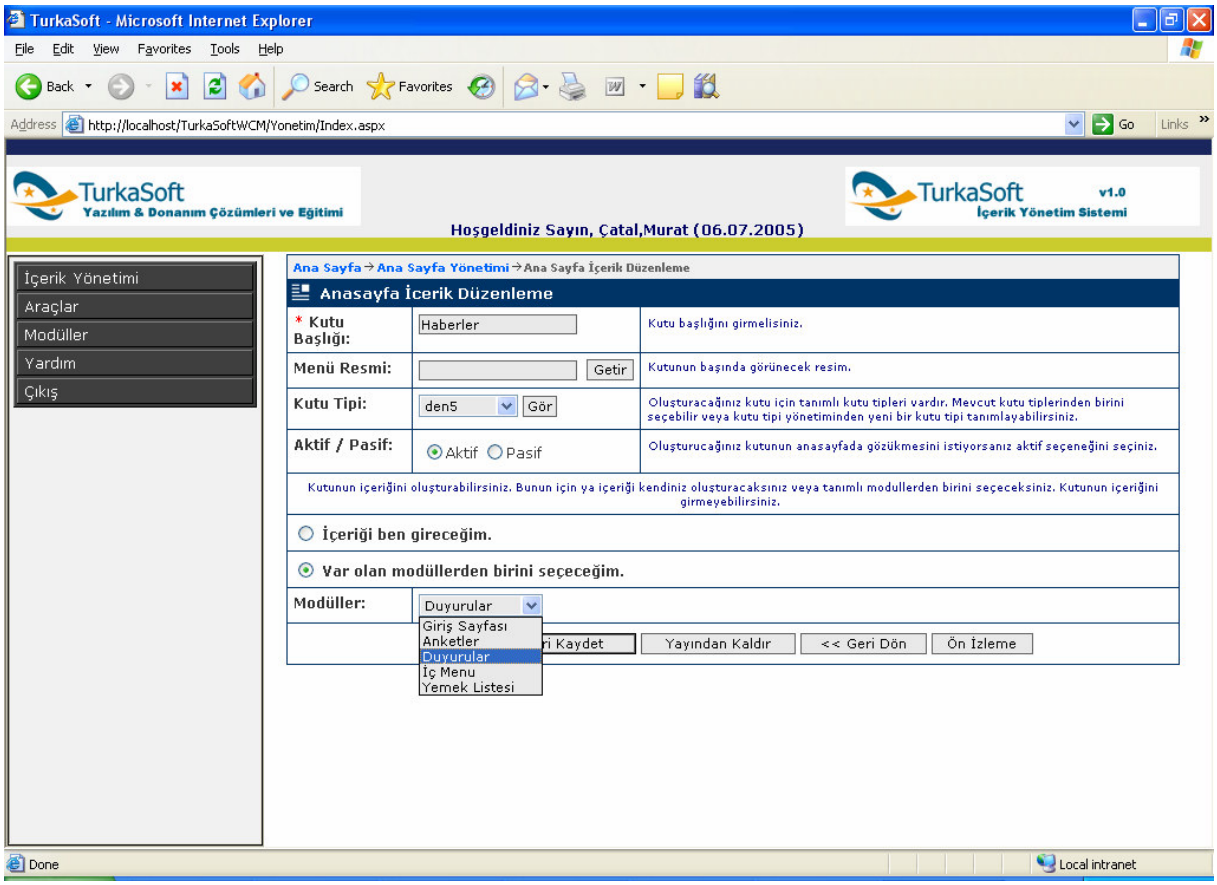


Figure 53 Homepage Content Management Page

Note: There are a lot of forms in project. Above, I only put some of them.

CONCLUSION

With this project it is aimed to administer and publish mid to large sized web sites, corporate and vertical portals online.

COMAS is built to administer and publish mid to large sized web sites, corporate and vertical portals. After the initial implementation by COMAS Admin, the site owners can easily manage, update and maintain the site.

COMAS has been developed with VB.Net and its powerful architecture. N-tier architecture has been used with presentation, process and database layers.

COMAS's modular structure allows development of custom components according to various needs. With the addition of new modules, the web site or the portal contains all the features and functions you need.

As would be explained in details, the above pages it is agreed to use ASP.NET for web programming language, C# and JavaScript for scripting language and Microsoft Office Access as being the database management system.

Above pages, I explained the advantages of using ASP.NET. At the beginning of project, I have already known these advantages, because of my experience. I only read a lot of articles and joint forums. At the end, I saw these advantages. I can advise everybody to use this platform in order to implement windows applications, web applications and web services.

REFERENCES

- **Çankaya , Mehmet Nuri:** ASP.NET, Seçkin Yayıncılık(2002)
- **Jerke, Noel:** Microsoft Office Access 2003 Professional Results, McGraw-Hill Osborne Media(2003)
- **Microsoft Corporation:** Microsoft .NET Framework SDK Help(2005)
- **Yıldırımoğlu , Murat:** Internet Information Server, Pusula Yayıncılık(2001)
- **Walther, Stephen:** ASP.NET Unleashed, Sams(2003)
- **Sceppa, David:** Microsoft ADO.NET, Microsoft ADO.NET(2002)
- **Walther, Stephen:**ASP.NET Kick Start, Sams(2003)
- **Taşdelen, Aykut:** ADO.NET, PUSULA Yayıncılık(2003)
- **Algan, Sefer:** .NET, Pusula Yayıncılık(2003)
- **C# Corner:** WWW-page, URL: www.csharp-corner.com(2004)
- **Microsoft Corporation:** WWW-page, URL: www.gotdotnet.com(2005)
- **DomainSpa LLC:** WWW-page, URL: www.csharpindex.com(2004)
- **Nedir?com Topluluğu:** WWW-page, URL: www.aspnedir.com(2005)
- **Jelsoft Enterprises Ltd.:** WWW-page, URL: www.programmersresource.com(2005)
- **Nedir?com Topluluğu:** WWW-page, URL: www.yazgelistir.com(2005)
- **Refsnes Data:** WWW-page, URL: www.w3schools.com(2005)
- **Microsoft Corporation:** WWW-page, URL: www.asp.net(2005)
- **CMP Media LLC:** WWW-page, URL: www.dotnetjunkies.com(2005)
- **DotNetBips:** WWW-page, URL: www.dotnetbips.com(2005)
- **Radore Hosting:** WWW-page, URL: www.maxiasp.net(2004)
- **Seeq:** WWW-page, URL: www.javasayfasi.com(2005)
- **.netWire:** WWW-page, URL: www.dotnetwire.com(2004)

- **ASPAlliance:** WWW-page, URL: www.aspalliance.com(2005)
- **SuperDotNet:** WWW-page, URL: www.superdotnet.com(2004)
- **CodeProject:** WWW-page, URL: www.codeproject.com(2005)
- **vbCity:** WWW-page, URL: www.vbcity.com(2005)
- **O'Reilly Media:** WWW-page, URL: www.ondotnet.com(2005)
- **Jupitermedia Corporation:** WWW-page, URL: www.codeguru.com(2005)
- **Open Source Technology Group:** WWW-page, URL: http://sourceforge.net(2005)
- **Dev Articles:** WWW-page, URL: www.devarticles.com(2005)
- **Jupitermedia Corporation:** WWW-page, URL: www.webreference.com(2005)
- **Matthew Reynolds:** WWW-page, URL: www.dotnet247.com(2004)
- **VB.NET Heaven:** WWW-page, URL: www.vbdotnetheaven.com(2003)
- **Mindcracker LLC:** WWW-page, URL: www.aspdotnetheaven.com(2005)
- **Nedir?com Topluluğu:** WWW-page, URL: www.csharpnedir.com(2005)
- **Bilgi Yönetimi:** WWW-page, URL: www.bilgiyonetimi.org(2002)
- **ASP Objects.com / CSTI:** WWW-page, URL: www.aspobjects.com(2005)
- **ASP Free Cluster 4:** WWW-page, URL: www.aspfree.com(2005)
- **ComponentSpace Pty Ltd.:** WWW-page, URL: www.componentspace.com(2005)
- **InfoGenius Inc.:** WWW-page, URL: www.411asp.net(2005)
- **ComponentSource:** WWW-page, URL: www.componentsource.com(2005)
- **Chilkat Software:** WWW-page, URL: www.example-code.com(2005)
- **DotNet Türk:** WWW-page, URL: www.dotnetturk.com(2004)
- **BOSSdev Inc.:** WWW-page, URL: www.visualasp.com(2005)
- **superdotnet.com:** WWW-page, URL: www.superdotnet.com(2004)
- **Brian Bischof:** WWW-page, URL: www.crystalreportsbook.com(2004)

- **KnowledgeStorm Inc.:** WWW-page, URL: http://contentmanagement.knowledgestorm.com(2005)
- **Creative Commons:** WWW-page, URL: http://www.cmswiki.com(2004)
- **Wikimedia Foundation:** WWW-page, URL: http://en.wikipedia.org(2005)
- **DotNetNuke:** WWW-page, URL: http://www.dotnetnuke.com(2005)
- **Microsoft Corporation:** WWW-page, URL: http://www.microsoft.com(2005)
- **İstanbul Kurumsal Gelişim:** WWW-page, URL: http://www.projeyonetimi.com (2004)
- **mutasyon.net:** WWW-page, URL: http://www.mutasyon.net(2004)
- **Nedir?com Topluluğu:** WWW-page, URL: http://www.yazilimmimarlari.com(2005)
- **Nedir?com Topluluğu:** WWW-page, URL: http://www.sqlnedir.com(2005)