

NEAR EAST UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING



HOTEL RESERVATION PROGRAM

- Otelim -

COM 400

GRADUATION PROJECT

By

Onur Gökmen

Supervisor: Ümit Soyer

June, 2006- Nicosia



TABLE OF CONTENTS

ACKNOWLEDGEMENTS.....	3
ABSTRACT.....	4
INTRODUCTION.....	6
CHAPTER I	7
1.1 SOFTWARE	7
1.2 VISUAL BASIC	7
1.3 INTERNET	8
1.3.1 BROWSER.....	11
1.3.2 HTML	12
1.3.3 WEBSITE	13
1.4 ASP	15
1.5 A BRIEF HISTORY OF DATABASES	15
1.5.1 Dr. Codd's 12 Rules for a Relational Database Model	15
1.6 SQL	19
1.7 ACTIVEX	21
CHAPTER II.....	23
STRUCTURE OF THE PROGRAM.....	23
2.1 TABLES	23
2.1.1 Users.....	23
2.1.2 Userlevels	24
2.1.3 Rooms.....	25
2.1.4 Roomtypes.....	Error! Bookmark not defined.
2.1.5 Reservations.....	27
2.2 THE TABLE RELATIONS	28
CHAPTER III	29
3.1 DEFAULT.ASP	29
3.2 SIGNIN.ASP	32
3.3 REGISTER.ASP	33
3.4 USERINFO.ASP	34
3.5 SIGNOUT.ASP	35
3.6 RESERVATIONS.ASP	36
3.7 RESERVEP.ASP.....	37
3.8 RES.ASP	38
3.9 CONTACTUS.ASP.....	39
CHAPTER IV	40
4.1 THE ADMINISTRATION OF THE WEBSITE (OTELIM.COM)	40
4.2 INCLUDED PAGES.....	43
4.2.1 Ust.asp.....	43
4.2.2 Alt.asp	44
4.2.3 Functions.asp.....	Error! Bookmark not defined.
4.2.4 Style.css	44
4.2.5 Rooms.asp	45
CONCLUSION	46
SUMMARY	46
RECOMMENDATION.....	46
REFERENCE.....	47
APENDIX 1 USER SIDE PAGES' CODES	48
DEFAULT.ASP	48
SIGNIN.ASP	49

SIGNINP.ASP	51
REGISTER.ASP	53
REGISTERP.ASP	57
SIGNOUT.ASP	60
FORGOTPASSWORD.ASP	61
FORGOTPASSWORDP.ASP	62
CONTACTUS.ASP	64
UPDATEINFO.ASP	65
UPDATEINFOP.ASP	70
RES.ASP	72
RESERVE.ASP	75
RESERVEP.ASP	77
 APENDIX 2 INCLUDED PAGES	 84
UST.ASP	84
ALT.ASP	88
ROOMS.ASP	89
FUNCTIONS.ASP	91
STYLE.CSS	95
 APENDIX 3 THE CODES FOR THE ADMINISTRATION PART	 100
ADMINISTRATOR_OF_THE_SITE.ASP	100
ADMINTOP.ASP	100
USERS.ASP	101
USERD.ASP	104
USERUP.ASP	110
RESERVATINOS.ASP	111
RESERVATINOSD.ASP	115
RESERVATIONSU.ASP	115
RESERVATIONSUP.ASP	117
ROOM.ASP	119
ROOMD.ASP	122
ROOMU.ASP	123
ROOMUP.ASP	126

ACKNOWLEDGEMENTS

I would like to extend my gratitude and thanks to Dr Suat İ. Günsel, Founding President of Near East University, for the opportunity he offered me, to make this website possible.

I am also grateful to Prof Dr Fahreddin Mehmedov, the dean of the engineering faculty for his motivation and encouragement which have been crucial; such motivation and encouragement have confirmed me in my view.

I must also thank Ümit Soyer, my project supervisor, who contributed greatly to my study with his constructive comments, suggestions and patient ,supervising throughout my study.

Finally, I wish to express my gratification and indebtedness to my teachers at the department of computer engineering, who provided me with valuable references which I really benefited from.

ABSTRACT

In this project, I have studied on **Hotel Reservation - Otelim**. Thus, at the very beginning of this section, it is reasonable to say that computer performs the tasks, such as electronic communication, under the control of a set of instructions called a program. Programming languages allow people to communicate with computers ; that is, people use computers in many ways. In commerce, computers track inventories with bar codes and scanners, check the credit status of customers, and transfer funds electronically.

Before going further, we should say that the physical computer and its components are known as **hardware**. Computer hardware includes the memory that stores data and program instructions, the central processing unit (CPU) that carries out program instructions; the input devices, such as a keyboard or mouse, that allow the user to communicate with the computer; the output devices, such as printers and video display monitors, that enable the computer to present information to the user, and *buses* (hardware lines or wires) that connect these and other computer components. The programs that run the computer are called **software**. Generally speaking, **software** is designed to perform a particular type of task- for example, to control commerce system and to facilitate both the formation of shopping for consumers and the commercial expansion as indicated in this study.

In order to carry out a program, computer programmers write code in languages known as **an assembly language** and **high-level languages** in use today which include **C, C⁺⁺, Pascal, delphi, COBOL, Visual basic , Java** and the likes.

What is more, some languages, such as the **markup languages** as **HTML** and their variants have been taken into consideration for the purpose of this project I have carried out. From the angle of the project, I should say that computers can communicate with other computers through a series of connections and associated hardware called a **network**. The advantage of a network is that data can be exchanged rapidly ,and

software and **hardware** resources , such as **hard-disk space** or **printers** can be shared. **Networks** also allow remote use of a computer by user.

Web Site, in computer science, is a file of information located on a server connected to the World Wide Web (WWW). The WWW is a set of protocols and software that allows the global computer network called the **Internet** to display multimedia documents. Websites may include text, photographs, illustrations as can be seen in this **graduation project**. Every web site has a specific address on the WWW, called a Uniform Resource Locator (URL). These addresses end in extensions that indicate the type of organization sponsoring the web site, for example, **.gov** for government agencies, **.edu** for academic institutions, and **.com** for commercial enterprises. **Our address** in the study is **WWW. Otelim Com**. The user's computer must be connected to the Internet and have a special software program called a browser to retrieve and read information from a web site.

To find and make use of this **website**, a user can consult an **Internet reference guide**. Note that **Websites** can easily be modified and updated, so the content of many sites changes frequently.

By taking into consideration the explanation given above, we can say that the main body of this **graduation project** has been studied under the chapters indicated below:

Chapter 1 is concerned with **Hotel Reservation**; chapter 2 focuses on the structure of **Hotel Reservation** – program module and data bases; chapter 3 puts great emphasis on the **development of Hotel Reservation**- flow charts. Software implementation is studied in chapter 4; **conclusion** has been given at the very end of this study; reference and appendix follow conclusion.

INTRODUCTION

The graduation project **Hotel Reservation - Otelim** has grown out of my hard and long study. The program which runs the computer is called **software**. The software has been designed to perform the task to control **Hotel reservation**, to reserve room by using internet. This **program** which I have carried out follows the same principles as traditional commerce realised in business life. We can easily say that in **E-commerce**, the buyer and the seller transact business over **network computer or the Internet**. To make purchases, the program which I have carefully designed and carried out offers buyers convenience. The program provides buyers with the opportunity to visit the **website** 24 hours a day and seven days a week to see and compare the prices and they may decide to make purchases, without having to leave their homes and offices.

Notice that the user's computer must be connected to the **internet** and have a special **software program** called **browser** to retrieve and read information from the **website** provided. The content on the website I have done contains hypertext and pictures that can also serve as links to the other pages or sites. By clicking on the **hypertext** or pictures with a mouse, users instruct their browser program to connect to the pages on the same website or to the other websites by the **URL** contained in the hypertext link. These links are embedded in the website through the use of **Hypertext Markup Language (HTML)**, a special language that encodes the links with the correct URL.

CHAPTER I

1.1 SOFTWARE

Software (computer), computer programs; instructions that cause the hardware—the machines—to do work. Software as a whole can be divided into a number of categories based on the types of work done by programs. The two primary software categories are operating systems (system software), which control the workings of the computer, and application software, which addresses the multitude of tasks for which people use computers. System software thus handles such essential, but often invisible, chores as maintaining disk files and managing the screen, whereas application software performs word processing, database management, and the like. Two additional categories that are neither system nor application software, although they contain elements of both, are network software, which enables groups of computers to communicate, and language software, which provides programmers with the tools they need to write programs..

In addition to these task-based categories, several types of software are described based on their method of distribution. These include the so-called canned programs or packaged software developed and sold primarily through retail outlets; freeware and public-domain software, which is made available without cost by its developer; shareware, which is similar to freeware but usually carries a small fee for those who like the program; and the infamous vapourware, which is software that either does not reach the market or appears much later than promised.

1.2 VISUAL BASIC

Hungarian-American mathematician John Kemeny and American mathematician Thomas Kurtz at Dartmouth College in Hanover, New Hampshire, developed BASIC (Beginner's All-purpose Symbolic Instruction Code) in 1964. The language was easier to learn than its predecessors and became popular due to its friendly, interactive nature and its inclusion on early personal computers. Unlike languages that require all their instructions to be translated into machine code first, BASIC is turned into machine language line by line as the program runs. BASIC commands typify high-level languages because of their simplicity and their closeness to natural human language. For example, a program that divides a number in half can be written as


```
0 INPUT "ENTER A NUMBER," X
```

```
20 Y=X/2
```

```
30 PRINT "HALF OF THAT NUMBER IS," Y
```

The numbers that precede each line are chosen by the programmer to indicate the sequence of the commands. The first line prints "ENTER A NUMBER" on the computer screen followed by a question mark to prompt the user to type in the number labeled "X." In the next line, that number is divided by two and stored as "Y." In the third line, the result of the operation is displayed on the computer screen. Even though BASIC is almost never used any more, this simple program demonstrates how data are stored and manipulated in most high-level programming languages.

1.3 INTERNET

Internet, computer-based global information system. The Internet is composed of many interconnected computer networks. Each network may link tens, hundreds, or even thousands of computers, enabling them to share information with one another and to share computational resources such as powerful supercomputers and databases of information. The Internet has made it possible for people all over the world to effectively and inexpensively communicate with one another. Unlike traditional broadcasting media, such as radio and television, the Internet does not have a centralized distribution system. Instead, an individual who has Internet access can communicate directly with anyone else on the Internet, make information available to others, find information provided by others, or sell products with a minimum overhead cost.

The Internet has brought new opportunities to government, business, and education. Governments use the Internet for internal communication, distribution of information, and automated tax processing. In addition to offering goods and services online to customers, businesses use the Internet to interact with other businesses. Many individuals use the Internet for shopping, paying bills, and online banking. Educational institutions use the Internet for research and to deliver courses to students at remote sites.

The Internet's success arises from its flexibility. Instead of restricting component networks to a particular manufacturer or particular type, Internet technology allows interconnection of any kind of computer network. No network is too large or too small, too fast or too slow to be interconnected. Thus, the Internet includes inexpensive networks that can only connect a few computers within a single room as well as expensive networks that can span a continent and connect thousands of computers. *See* Local Area Network.

Internet service providers (ISPs) provide Internet access to customers for a monthly fee. A customer who subscribes to an ISP's service uses the ISP's network to access the Internet. Because ISPs offer their services to the general public, the networks they operate are known as public access networks. In the United States, as in many countries, ISPs are private companies; in countries where telephone service is a government-regulated monopoly, the government often controls ISPs.

An organization that has many computers usually owns and operates a private network, called an intranet, that connects all the computers within the organization. To provide Internet service, the organization connects its intranet to the Internet. Unlike public access networks, intranets are restricted to provide security. Only authorized computers at the organization can connect to the intranet, and the organization restricts communication between the intranet and the global Internet. The restrictions allow computers inside the organization to exchange information but keep the information confidential and protected from outsiders.

The Internet has grown tremendously since its inception, doubling in size every 9 to 14 months. In 1981 only 213 computers were connected to the Internet. By 2000 the number had grown to more than 100 million. The current number of people who use the Internet can only be estimated. One survey found that there were 61 million Internet users worldwide at the end of 1996, 148 million at the end of 1998, and 407 million by the end of 2000. Some analysts estimate that the number of users will double again by the end of 2002. From its inception in the 1970s until the late 1980s the Internet was a U.S. government-funded communication and research tool restricted almost exclusively to academic and military uses. As government restrictions were lifted in the early 1990s, the Internet became commercial. In 1995 the World Wide Web (WWW) replaced file transfer as the application used for most Internet traffic. Companies, individuals, and

institutions use the Internet in many ways. Companies use the Internet for electronic commerce, also called **e-commerce**, including advertising, selling, buying, distributing products, and providing customer service. In addition, companies use the Internet for business-to-business transactions, such as exchanging financial information and accessing complex databases. Businesses and institutions use the Internet for voice and video conferencing and other forms of communication that enable people to *telecommute* (work away from the office using a computer). The use of electronic mail (e-mail) speeds communication between companies, among coworkers, and among other individuals. Media and entertainment companies use the Internet for online news and weather services and to broadcast audio and video, including live radio and television programs. Online chat allows people to carry on discussions using written text. The term *Internet access* refers to the communication between a residence or a business and an ISP that connects to the Internet. Access falls into two broad categories: dedicated and dial-up. With dedicated access, a subscriber's computer remains directly connected to the Internet at all times by a permanent, physical connection. Most large businesses have high-capacity dedicated connections; small businesses or individuals who desire dedicated access choose technologies such as digital subscriber line (DSL) or cable modems, which both use existing wiring to lower cost. A DSL sends data across the same wires that telephone service uses, and cable modems use the same wiring that cable television uses. In each case, the electronic devices that are used to send data over the wires employ separate frequencies or channels that do not interfere with other signals on the wires. Thus, a DSL Internet connection can send data over a pair of wires at the same time the wires are being used for a telephone call, and cable modems can send data over a cable at the same time the cable is being used to receive television signals. The user usually pays a fixed monthly fee for a dedicated connection. In exchange, the company providing the connection agrees to relay data between the user's computer and the Internet.

Dial-up is the least expensive access technology, but it is also the least convenient. To use dial-up access, a subscriber must have a telephone modem, a device that connects a computer to the telephone system and is capable of converting data into sounds and sounds back into data. The user's ISP provides software that controls the modem. To access the Internet, the user opens the software application, which causes the dial-up modem to place a toll-free telephone call to the ISP. A modem at the ISP answers the

call, and the two modems use audible tones to send data in both directions. When one of the modems is given data to send, the modem converts the data from the digital values used by computers—numbers stored as a sequence of 1s and 0s—into tones. The receiving side converts the tones back into digital values. Unlike dedicated access technologies, a dial-up modem does not use separate frequencies, so the telephone line cannot be used for regular Research on dividing information into packets and switching them from computer to computer began in the 1960s. The U.S. Department of Defense Advanced Research Projects Agency (ARPA) funded a telephone calls at the same time a dial-up modem is sending data.

1.3.1 BROWSER

Browser, in computer science, a program that enables a computer to locate, download, and display documents containing text, sound, video, graphics, animation, and photographs located on computer networks. The act of viewing and moving about between documents on computer networks is called browsing. Users browse through documents on open, public-access networks called internets, or on closed networks called intranets. The largest open network is the Internet, a worldwide computer network that provides access to sites on the World Wide Web (WWW, the Web).

Browsers allow users to access Web information by locating documents on remote computers that function as Web servers. A browser downloads information over phone lines to a user's computer through the user's modem and then displays the information on the computer. Most browsers can display a variety of text and graphics that may be integrated into such a document, including animation, audio and video. Examples of browsers are Netscape, Internet Explorer, and Mosaic.

Browsers can create the illusion of traveling to an actual location in virtual space (hyperspace) where the document being viewed exists. This virtual location in hyperspace is referred to as a node, or a Web site. The process of virtual travel between Web sites is called navigating.

Documents on networks are called hypertext if the media is text only, or hypermedia if the media includes graphics as well as text. Every hypertext or hypermedia document on an internet has a unique address called a uniform resource locator (URL). Hypertext

documents usually contain references to other URLs that appear in bold, underlined, or colored text. The user can connect to the site indicated by the URL by clicking on it. This use of a URL within a Web site is known as a hyperlink. When the user clicks on a hyperlink, the browser moves to this next server and downloads and displays the document targeted by the link. Using this method, browsers can rapidly take users back and forth between different sites.

Common features found in browsers include the ability to automatically designate a Web site to which the browser opens with each use, the option to create directories of favorite or useful Web sites, access to search engines (programs that permit the use of key words to locate information on the Internet, an internet or an intranet), and the ability to screen out certain types of information by blocking access to certain categories of sites.

A browser's performance depends upon the speed and efficiency of the user's computer, the type of modem being used, and the bandwidth of the data-transmission medium (the amount of information that can be transmitted per second). Low bandwidth results in slow movement of data between source and recipient, leading to longer transmission times for documents. Browsers may also have difficulty reaching a site during times of heavy traffic on the network or because of high use of the site.

The most commonly used browsers for the Web are available for free or for a small charge and can be downloaded from the Internet. Browsers have become one of the most important tools—ranking with e-mail—for computer network users. They have provided tens of millions of people with a gateway to information and communication through the Internet.

1.3.2 HTML

Hypertext Markup Language (HTML) in computer science, the standard text-formatting language since 1989 for documents on the interconnected computing network known as the World Wide Web. HTML documents are text files that contain two parts: content that is meant to be rendered on a computer screen; and markup or tags, encoded

information that directs the text format on the screen and is generally hidden from the user. HTML is a subset of a broader language called Standard Generalized Markup Language (SGML), which is a system for encoding and formatting documents, whether for output to a computer screen or to paper.

Some tags in an HTML document determine the way certain text, such as titles, will be formatted. Other tags cue the computer to respond to the user's actions on the keyboard or mouse. For instance, the user might click on an icon (a picture that represents a specific command), and that action might call another piece of software to display a graphic, play a recording, or run a short movie. Another important tag is a link, which may contain the Uniform Resource Locator (URL) of another document. The URL can be compared to an address where a particular document resides. The document may be stored on the same computer as the parent document or on any computer connected to the World Wide Web. The user can navigate from document to document simply by clicking on these links. HTML also includes markups for forms, that let the user fill out information and electronically send, or e-mail, the data to the document author, initiate sophisticated searches of information on the Internet, or order goods and services.

The software that permits the user to navigate the World Wide Web and view HTML-encoded documents is called a browser. It interprets the HTML tags in a document and formats the content for screen display. Since HTML is an accepted standard, anyone can build a browser without concerning themselves with what form various documents will assume, unlike documents produced by typical word processors, which must be translated into a different format if another word processing application is used. Most sites on the World Wide Web adhere to HTML standards and, because HTML is easy to use, the World Wide Web has grown rapidly. HTML continues to evolve, however, so browsers must be upgraded regularly to meet the revised standards.

1.3.3 WEBSITE

Web Site, in computer science, file of information located on a server connected to the World Wide Web (WWW). The WWW is a set of protocols and software that allows the global computer network called the Internet to display multimedia documents. Web

sites may include text, photographs, illustrations, video, music, or computer programs. They also often include links to other sites in the form of hypertext, highlighted or colored text that the user can click on with their mouse, instructing their computer to jump to the new site.

Every web site has a specific address on the WWW, called a Uniform Resource Locator (URL). These addresses end in extensions that indicate the type of organization sponsoring the web site, for example, .gov for government agencies, .edu for academic institutions, and .com for commercial enterprises. The user's computer must be connected to the Internet and have a special software program called a browser to retrieve and read information from a web site. Examples of browsers include Navigator from the Netscape Communications Corporation and Explorer from the Microsoft Corporation.

The content presented on a web site usually contains hypertext and icons, pictures that also serve as links to other sites. By clicking on the hypertext or icons with their mouse, users instruct their browser program to connect to the web site specified by the URL contained in the hypertext link. These links are embedded in the web site through the use of Hypertext Markup Language (HTML), a special language that encodes the links with the correct URL.

Web sites generally offer an appearance that resembles the graphical user interfaces (GUI) of Microsoft's Windows operating system, Apple's Macintosh operating system, and other graphics based operating systems. They may include scroll bars, menus, buttons, icons, and toolbars, all of which can be activated by a mouse or other input device.

To find a web site, a user can consult an Internet reference guide or directory, or use one of the many freely available search engines, such as WebCrawler from America Online Incorporated. These engines are search and retrieval programs, of varying sophistication, that ask the user to fill out a form before executing a search of the WWW for the requested information. The user can also create a list of the URLs of frequently visited web sites. Such a list helps a user recall a URL and easily access the desired web site. Web sites are easily modified and updated, so the content of many sites changes **frequently**.

1.4 ASP

A technology pioneered by the Microsoft Corporation which triggers processing at a WEB SERVER when a WEB PAGE is requested by a BROWSER. Normally when a browser requests a page from a WEB SERVER it is sent directly back to the browser. Active Server Page technology triggers some processing, such as inserting topical data, when a Web page which has the FILE EXTENSION ASP is requested. The server examines the file, identifies any SCRIPTs that need to be executed, executes them and inserts the results into the Web page. A typical example of an application that might use ASP technology is network monitoring, where pages which have dynamic information embedded in them that describes the current state of a network can be sent back to a SYSTEM ADMINISTRATOR. This technology is similar in intent to that of SERVER SIDE INCLUDEs. It is usually referred to as ASP.

1.5 A Brief History of Databases

A little background on the evolution of databases and database theory will help you understand the workings of SQL. Database systems store information in every conceivable business environment. From large tracking databases such as airline reservation systems to a child's baseball card collection, database systems store and distribute the data that we depend on. Until the last few years, large database systems could be run only on large mainframe computers. These machines have traditionally been expensive to design, purchase, and maintain. However, today's generation of powerful, inexpensive workstation computers enables programmers to design software that maintains and distributes data quickly and inexpensively.

1.5.1 Dr. Codd's 12 Rules for a Relational Database Model

The most popular data storage model is the relational database, which grew from the seminal paper "A Relational Model of Data for Large Shared Data Banks," written by Dr. E. F. Codd in 1970. SQL evolved to service the concepts of the relational database

model. Dr. Codd defined 13 rules, oddly enough referred to as Codd's 12 Rules, for the relational model:

0. A relational DBMS must be able to manage databases entirely through its relational capabilities.

1. Information rule-- All information in a relational database (including table and column names) is represented explicitly as values in tables.

2. Guaranteed access--Every value in a relational database is guaranteed to be accessible by using a combination of the table name, primary key value, and column name.

3. Systematic null value support--The DBMS provides systematic support for the treatment of null values (unknown or inapplicable data), distinct from default values, and independent of any domain.

4. Active, online relational catalog--The description of the database and its contents is represented at the logical level as tables and can therefore be queried using the database language.

5. Comprehensive data sublanguage--At least one supported language must have a well-defined syntax and be comprehensive. It must support data definition, manipulation, integrity rules, authorization, and transactions.

6. View updating rule--All views that are theoretically updatable can be updated through the system.

7. Set-level insertion, update, and deletion--The DBMS supports not only set-level retrievals but also set-level inserts, updates, and deletes.

8. Physical data independence--Application programs and ad hoc programs are logically unaffected when physical access methods or storage structures are altered.

9. Logical data independence--Application programs and ad hoc programs are logically unaffected, to the extent possible, when changes are made to the table structures.

10. Integrity independence--The database language must be capable of defining integrity rules. They must be stored in the online catalog, and they cannot be bypassed.

11. Distribution independence--Application programs and ad hoc requests are logically unaffected when data is first distributed or when it is redistributed.

12. Nonsubversion--It must not be possible to bypass the integrity rules defined through the database language by using lower-level languages.

Most databases have had a "parent/child" relationship; that is, a parent node would contain file pointers to its children.

This method has several advantages and many disadvantages. In its favor is the fact that the physical structure of data on a disk becomes unimportant. The programmer simply stores pointers to the next location, so data can be accessed in this manner. Also, data can be added and deleted easily. However, different groups of information could not be easily joined to form new information. The format of the data on the disk could not be arbitrarily changed after the database was created. Doing so would require the creation of a new database structure.

Codd's idea for an RDBMS uses the mathematical concepts of relational algebra to break down data into sets and related common subsets.

Because information can naturally be grouped into distinct sets, Dr. Codd organized his database system around this concept. Under the relational model, data is separated into sets that resemble a table structure. This table structure consists of individual data elements called columns or fields. A single set of a group of fields is known as a record or row. For instance, to create a relational database consisting of employee data, you might start with a table called `EMPLOYEE` that contains the following pieces of information: `Name`, `Age`, and `occupation`. These three pieces of data make up the fields in the `EMPLOYEE` table, shown in Table 1.1.

Table 1.1. The EMPLOYEE table.

Name	Age	Occupation
Will Williams	25	Electrical engineer
Dave Davidson	34	Museum curator
Jan Janis	42	Chef
Bill Jackson	19	Student
Don DeMarco	32	Game Programmer
'Becky Boudreaux	25	Model

The six rows are the records in the `EMPLOYEE` table. To retrieve a specific record from this table, for example, Dave Davidson, a user would instruct the database management system to retrieve the records where the `NAME` field was equal to Dave Davidson. If the DBMS had been instructed to retrieve all the fields in the record, the employee's name, age, and occupation would be returned to the user. SQL is the language that tells the database to retrieve this data. A sample SQL statement that makes this query is

Select * FROM EMPLOYEE

The various data items can be grouped according to obvious relationships (such as the relationship of `Employee Name` to `Employee Age`), the relational database model gives the database designer a great deal of flexibility to describe the relationships between the data elements. Through the mathematical concepts of join and union, relational databases can quickly retrieve pieces of data from different sets (tables) and return them to the user or program as one "joined" collection of data. The join feature enables the designer to store sets of information in separate tables to reduce repetition.

Here's a simple example that shows how data can be logically divided between two tables. Table 1.2 is called `RESPONSIBILITIES` and contains two fields: `NAME` and `DUTIES`.

Table 1.2. The RESPONSIBILITIES table.

Name	Duties
Becky Boudreaux	Smile
Becky Boudreaux	Walk
Bill Jackson	Study
Bill Jackson	Interview for jobs

It would be improper to duplicate the employee's AGE and OCCUPATION fields for each record. Over time, unnecessary duplication of data would waste a great deal of hard disk space and increase access time for the RDBMS. However, if NAME and DUTIES were stored in a separate table named RESPONSIBILITIES, the user could join the RESPONSIBILITIES and EMPLOYEE tables on the NAME field. Instructing the RDBMS to retrieve all fields from the RESPONSIBILITIES and EMPLOYEE tables where the NAME field equals Becky Boudreaux would return Table 1.3.

Table 1.3. Return values from retrieval where NAME equals Becky Boudreaux.

Name	Age	Occupation	Duties
Becky Boudreaux	25	Model	Smile
Becky Boudreaux	25	Model	Walk

1.6 SQL

Finally, Electronic Commerce or e-commerce, the exchange of goods and services by means of the Internet or other computer networks. E-commerce follows the same basic principles as traditional commerce—that is, buyers and sellers come together to exchange goods for money. But rather than conducting business in the traditional way—in stores and other “brick and mortar” buildings or through mail order catalogs and telephone operators—in e-commerce buyers and sellers transact business over networked computers.

E-commerce offers buyers convenience. They can visit the World Wide Web sites of multiple vendors 24 hours a day and seven days a week to compare prices and make purchases, without having to leave their homes or offices. In some cases, consumers can immediately obtain a product or service, such as an electronic book, a music file, or computer software, by downloading it over the Internet.

For sellers, e-commerce offers a way to cut costs and expand their markets. They do not need to build, staff, or maintain a store or print and distribute mail order catalogs. Automated order tracking and billing systems cut additional labour costs, and if the product or service can be downloaded, e-commerce firms have no

Browser, in computer science, a program that enables a computer to locate, download, and display documents containing text, sound, video, graphics, animation, and photographs located on computer networks. The act of viewing and moving about between documents on computer networks is called browsing. Users browse through documents on open, public-access networks called internets, or on closed networks called intranets. The largest open network is the Internet, a worldwide computer network that provides access to sites on the World Wide Web (WWW, the Web).

Browsers allow users to access Web information by locating documents on remote computers that function as Web servers. A browser downloads information over phone lines to a user's computer through the user's modem and then displays the information on the computer. Most browsers can display a variety of text and graphics that may be integrated into such a document, including animation, audio and video. Examples of browsers are Netscape, Internet Explorer, and Mosaic.

Browsers can create the illusion of travelling to an actual location in virtual space (hyperspace) where the document being viewed exists. This virtual location in hyperspace is referred to as a node, or a Web site. The process of virtual travel between Web sites is called navigating.

Documents on networks are called hypertext if the media is text only, or hypermedia if the media includes graphics as well as text. Every hypertext or hypermedia document on an internet has a unique address called a uniform resource locator (URL). Hypertext documents usually contain references to other URLs that appear in bold, underlined, or

colored text. The user can connect to the site indicated by the URL by clicking on it. This use of a URL within a Web site is known as a hyperlink. When the user clicks on a hyperlink, the browser moves to this next server and downloads and displays the document targeted by the link. Using this method, browsers can rapidly take users back and forth between different sites.

Common features found in browsers include the ability to automatically designate a Web site to which the browser opens with each use, the option to create directories of favourite or useful Web sites, access to search engines (programs that permit the use of key words to locate information on the Internet, an internet or an intranet), and the ability to screen out certain types of information by blocking access to certain categories of sites.

A browser's performance depends upon the speed and efficiency of the user's computer, the type of modem being used, and the bandwidth of the data-transmission medium (the amount of information that can be transmitted per second). Low bandwidth results in slow movement of data between source and recipient, leading to longer transmission times for documents. Browsers may also have difficulty reaching a site during times of heavy traffic on the network or because of high use of the site.

The most commonly used browsers for the Web are available for free or for a small charge and can be downloaded from the Internet. Browsers have become one of the most important tools—ranking with e-mail—for computer network users. They have provided tens of millions of people with a gateway to information and communication through the Internet.

1.7 ActiveX

ActiveX is the name of a group of software technologies and methods that promise to make it easy to create "traditional business applications" with powerful Internet connectivity and "traditional Internet communications programs" with true computational power. Building on the OLE (Object Linking and Embedding) technology that has been available for some time, ActiveX essentially expands the realm of "object sharing" from merely the desktop to the whole Internet, making this new class of program possible. Since ActiveX technology is modular in design,

programs can be written in such a way that they can work as standalone applications, embedded "smart" objects within Visual Basic programs or Web pages, or as traditional OLE objects within business documents, all with the potential to communicate with the Internet, should you choose.

As ActiveX gains wider acceptance, the Internet will find itself changing from merely a wellspring of static information "pages" to a source of working programs that manipulate, display, change, or create new information or data. Communication with others through the Internet will occur within our word processors, spreadsheets, drawing programs, and so forth, without having to switch to dedicated browsers or Internet programs.

Or at least, that's the theory.

Being a very new technology, ActiveX has yet to take the Internet world by storm and change it overnight. But from the ActiveX technologies that have been exploited, it looks like Microsoft's vision of tomorrow's Internet might not be too far off. That's what this book, *Presenting ActiveX*, hopes to help you accomplish: to explore the possibilities ActiveX makes available to software developers.

As an introduction to ActiveX, this book will benefit the following people:

- Managers and planners interested in seeing what ActiveX technology will do for their products or organization.
- Web page developers looking to increase the appeal and functionality of their Web pages.
- Programmers who want to get an idea of how ActiveX works.
- Hobbyists who want to stay abreast of the latest Internet technologies.

With the certain concepts being explained above, we can easily move to chapter two, which is concerned with the **Structure Hotel Reservation**.

CHAPTER II

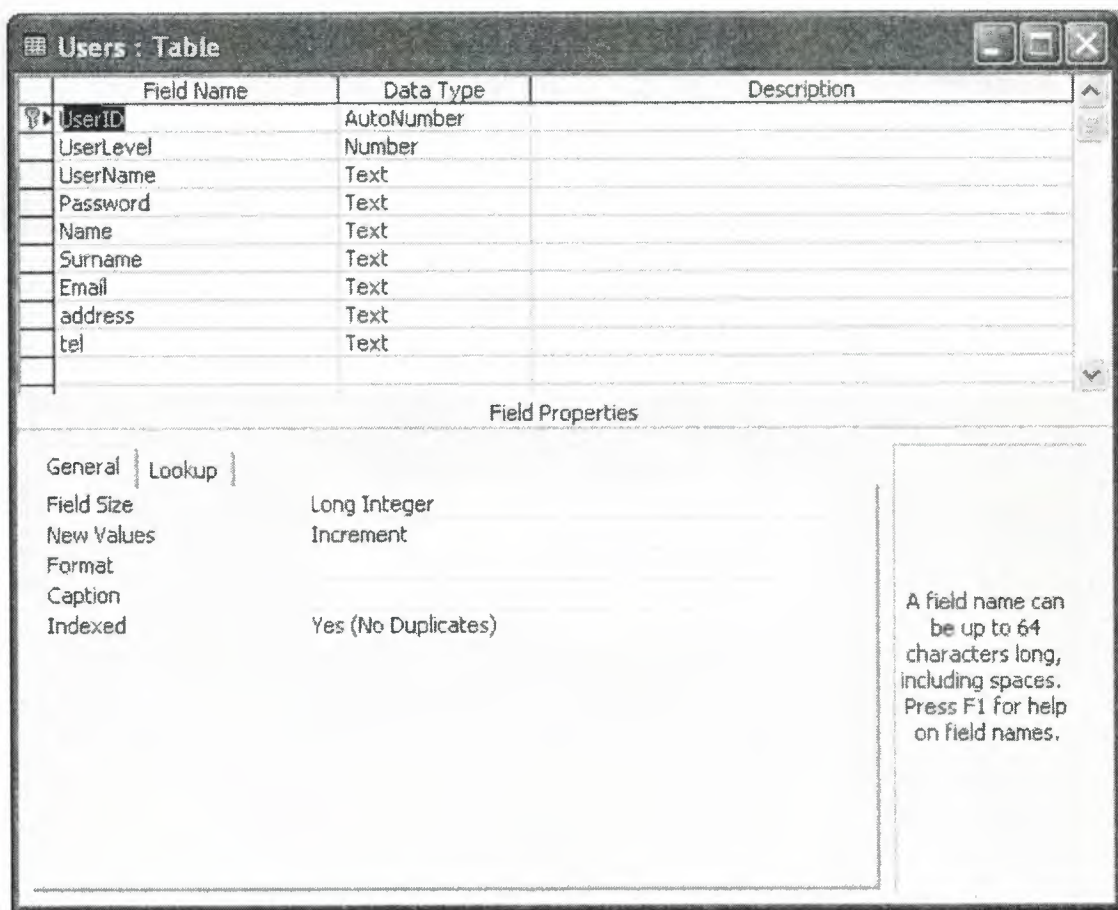
STRUCTURE OF THE PROGRAM

2.1 TABLES

The name of the database is reservation.mdb inside the db folder in this project. There are different tables for different purposes. I will explain all the tables one by one after that I will give the explanation for the relations.

2.1.1 Users

This table is designed to store the users' information. It includes these fields UserID, Userlevel, Username, Passwords, Name, Surname, E_mail, Address, Tel. The variable types of the fields is given below (Figure 2.1.1).



Field Name	Data Type	Description
UserID	AutoNumber	
UserLevel	Number	
UserName	Text	
Password	Text	
Name	Text	
Surname	Text	
Email	Text	
address	Text	
tel	Text	

Field Properties	
General	Lookup
Field Size	Long Integer
New Values	Increment
Format	
Caption	
Indexed	Yes (No Duplicates)

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Figure 2.1.1(Users table)

2.1.2 Userlevels

The userleveltbl is for storing userlevels' (Figure 2.1.2). The fields and their types are given in the figure.

Field Name	Data Type	Description
UserLevelId	AutoNumber	
UserLevelName	Text	

Field Properties	
General	Lookup
Field Size	Long Integer
New Values	Increment
Format	
Caption	
Indexed	Yes (No Duplicates)

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Figure 2.1.2(Userlevels table)

2.1.3 Rooms

This table is for storing the rooms in the hotel (Figure 2.1.3).

Rooms : Table

Field Name	Data Type	Description
RoomID	Number	
Stair	Text	
RoomTypeID	Number	

Field Properties

General | Lookup

Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	0
Validation Rule	
Validation Text	
Required	No
Indexed	Yes (No Duplicates)

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Figure 2.1.3(Rooms table)

2.1.4 Roomtypes

This table is for storing the room types (Figure 2.1.4).

Field Name	Data Type	Description
RoomTypeID	AutoNumber	
RoomType	Text	
BigPicture	Text	
Picture	Text	
Info	Text	
LongInfo	Memo	
Price	Number	

Field Properties

General | Lookup

Field Size: Long Integer
New Values: Increment
Format:
Caption:
Indexed: Yes (No Duplicates)

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Figure 2.1.4(Roomtypes table)

2.1.5 Reservations

This table is designed to normalize the reservations table. The information in this table was going to be saved for each reservations in one order so it is better to take the repeated information to another table (Figure 2.1.5).

Field Name	Data Type	Description
ReservationID	AutoNumber	
RoomID	Number	
UserID	Number	
NoofAdults	Number	
NoofChildren	Number	
Total	Number	
StartDate	Date/Time	
FinishDate	Date/Time	
ReservationDate	Date/Time	
ReservationTime	Date/Time	

Field Properties

General | Lookup

Field Size: Long Integer
New Values: Increment
Format:
Caption:
Indexed: Yes (No Duplicates)

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Figure 2.1.5 (Reservations table)

2.2 The Table Relations

The table relations are necessary in a database to make the design more easier. In this project in order to avoid to have redundant data and to make the searches easy. This process is called normalization. The relations are given below with a figure (Figure 2.2).

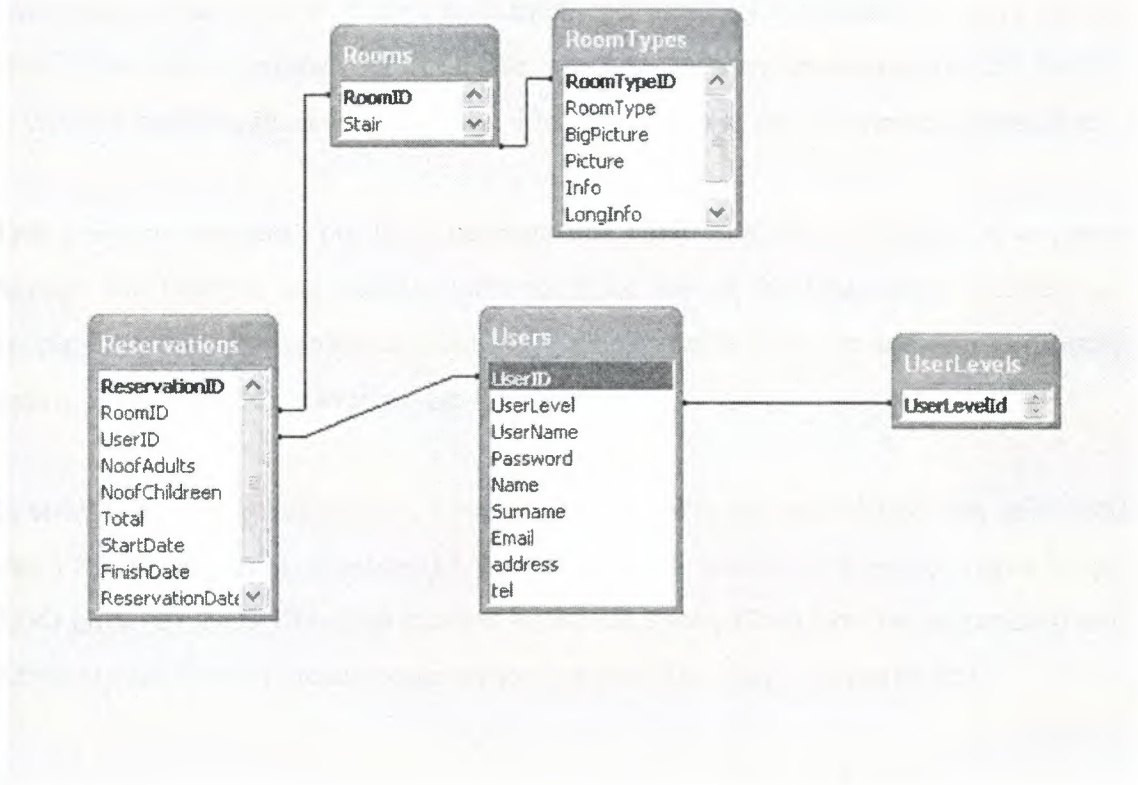


Figure 2.2 Table Relations

CHAPTER III

DEVELOPMENT OF THE PROGRAM

To design my graduation project and make it possible, I have used the network technology; what is more, I have used Internet technology specifically to carry out the Hotel Reservation program for Hotel Reservations vacancy Otelim.com-in this study, to conduct business functions remotely, I have made great use of Internet technology.

This program provides both the customers and hotel with the facilities of reservation through the Internet and enables them to make use of the reservation facilities; so people have the opportunities to visit our website 'otelim.com'. In addition to offering rooms, services online is available to the customers.

In order to carry out this project, a composition of different technologies has been used like VB (Visual Basic), Database (Access Database is used in this project), Java Script, ADO (ActiveX Data Object to connect to the database), Flash (for the animations) and different variations of picture compression formats (Gif, Jpeg, Animated gif).

3.1 Default.asp

Before going further I want to give a brief explanation about the **include** command. This command is used to divide pages to sub pages in order to make the design easy. The sub pages are designed to do some specific jobs (like checking user account). Instead of writing the same code to all the pages, you can include the page that does this job.

In this project the pages that are to be included are in the **Includes** folder. I will give certain explanation for all pages included later in this study.

The Default.asp is the main page of the site. It is formed from many pages. This page covers the following:

1.ust.asp : The page that is included at the top of the main page (all the pages include this page).

2.Adovbs.inc: To use the Record set object's different methods.

3.rooms.asp: To list the rooms.

4.alt.asp:Bottom of the page. It shows copyright information.

At the top of the main page there are links such as Login, User Details, Logout, E-mail, Search, Shopping cart, Services, Links, and Info About the Company (these links will be explained later). In the middle of the page there are the list of rooms. At last, at the bottom of the main page there is information about the copyright of my web page.

The links at the top provide users with the opportunity to change the user information, to login an existing account, to create a user account, to logout, to see their reservations made.

Everybody can explore all the pages(except admin pages) but in order to reserve a room he/she has to be a member.

The links at the top are used to navigate between the pages. The sign in link is to go to sign in page, the rooms link is used to turn back to the list of rooms, the contact us link is to see the contact information of our company, sign out is to sign out you're your account, update user info link is to update the information belongs to you, reservations link is to see the reservations made.(Figure3.1)

,



Room Types



Standard Maximum Occupation 3

240\$/day

[Reserve](#)

Standard Room-2 Twins-balcony-AC-sea view in some rooms-minibar-37 sq meters-plasma tv-safe

Standard Room has 2 Twin beds or a King. Rooms are 37 meters squared. Some rooms have a sea view. All rooms have balcony, safe, air conditioning, minibar, 28 inch plasma TV, and hairdryer. Children under 7 stay free in the room with parents and in existing bedding. Children age 7-12 receive a 50 percent reduction.



Family Suite (max 4)

731\$/day

[Reserve](#)

Suite-1 King and pullout-sea view-balcony-AC- sitting area-2 plasma TV-safe-hairdryer

Suite has 1 King and a pull-out sofa. All suites have a sea view, balcony, 1 1/2 bathrooms, and a sitting area. Suite has AC, minibar, 2- 28 inch plasma TVs, safe and hairdryer. Children under 7 stay free in room with parents and in existing bedding. Children 7-12 receive a 50 percent reduction.

Figure 3.1 (Default.asp)

3.2 signin.asp

This page is for registered users to login their account. The user writes his/her username and then when he/she presses the login button the signinp.asp page will check the username and password from the database and if both are correct the user will be directed to Default.asp. If one of the textboxes is empty or the username and/or password is wrong the user can not login to the account required and a message indicates that 'Username and/or password is wrong'. If the person is not a member there is a link "Click here to be a member" to be a member (Figure 3.2). Otherwise the user can successfully complete the login operation and the sessions will be written for the site to remember who he/she is.

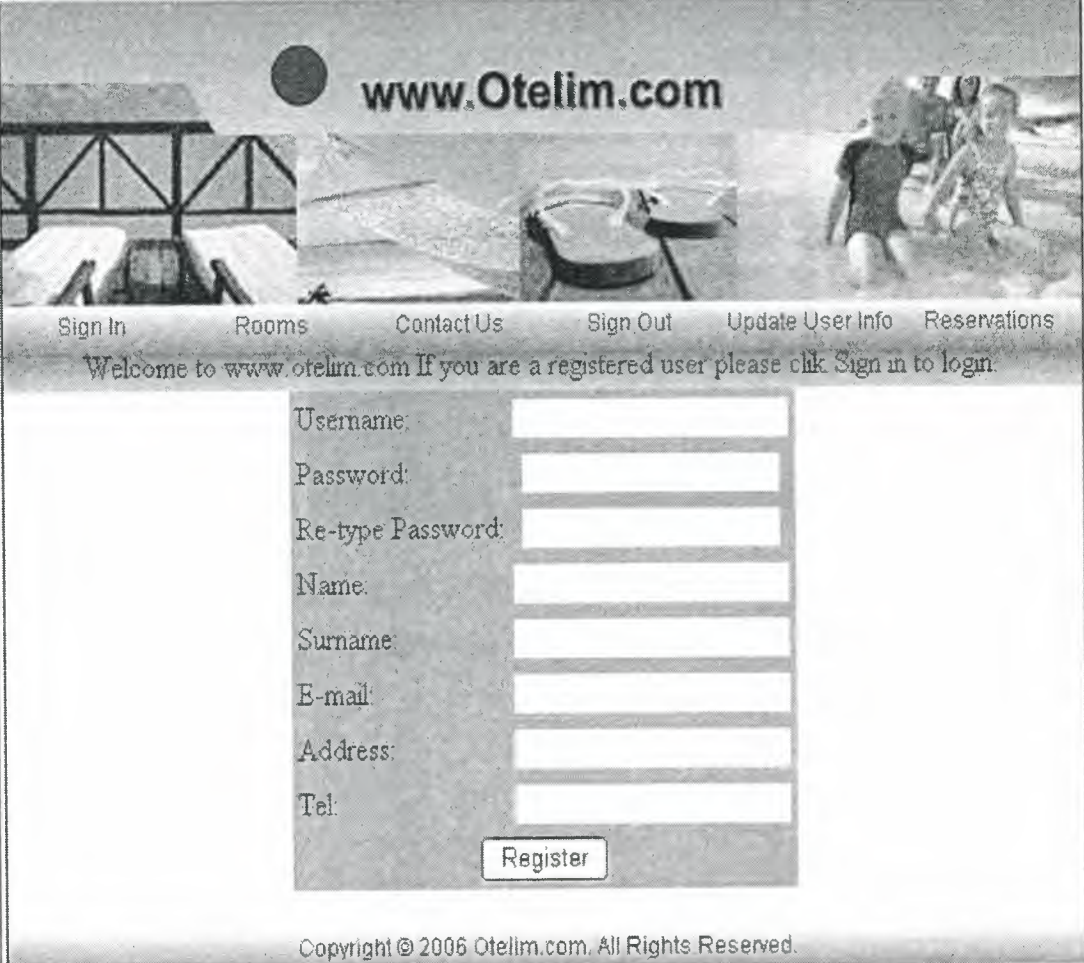


The screenshot shows the login page for www.Otelim.com. At the top, there is a header with the website name and a navigation menu with links: Sign In, Rooms, Contact Us, Sign Out, Update User Info, and Reservations. Below the menu, a welcome message reads: "Welcome to www.otelim.com If you are a registered user please click Sign in to login". The main content area contains a login form with two input fields: "Username:" and "Password:". Below these fields are three links: "New User", "Forgot Password", and a "Sign In" button. At the bottom of the page, there is a copyright notice: "Copyright © 2006 Otelim.com. All Rights Reserved".

Figure 3.2(Signin.asp)

3.3 Register.asp

The Register.asp page is for new users to create an account. Firstly the user has to fill the form. The page will check the information and will warn the user if there is any mistake. The Register.asp firstly checks the username if the username belongs to another user there will be a message 'The username that you have entered exists. Choose another and try again.'. After that, the page checks the passwords if they are not equal, that means the user has made a mistake; that's why there will be a message 'The passwords are not equal'. Throughout all these processes the user does not have to write all the information again and again because the page sends all the information back to the form (Figure 3.3).



www.Otelim.com

Sign In Rooms Contact Us Sign Out Update User Info Reservations

Welcome to www.otelim.com If you are a registered user please click Sign in to login.

Username:

Password:

Re-type Password:

Name:

Surname:

E-mail:

Address:

Tel:

Register

Copyright © 2006 Otelim.com. All Rights Reserved.

Figure3.3 (Register.asp)

3.4 updateinfo.asp

The user can not change the username but all the other information can be changed. This page acts like register page and checks the same things. If the information is valid the page updates that user's information (Figure 3.4).

www.Otelim.com

Sign In Rooms Contact Us Sign Out Update User Info Reservations

Welcome back Onur. Click on rooms for reservation

Admin Page

Username:

Password:

Re-type Password:

Name:

Surname:

E-mail:

Address:

Tel:

Copyright © 2006 Otelim.com. All Rights Reserved.

Figure3.4 (Updateinfo.asp)

3.5 Signout.asp

This page is for the users who logged in with an existing account. This page resets the session that is written in the sign in page and makes the site secure for the user. The figure for this page is given below.



Figure 3.5 (Signout.asp)

3.6 Reserve.asp

This page is for room reservation. The user enters from date and to date as date and number of adults and number of children. This page posts the information taken from the form to the reservep.asp

www.Otelim.com

Sign In Rooms Contact Us Sign Out Update User Info Reservations

Welcome back Onur. Click on rooms for reservation

Admin Page

From Date: Jun 24 2006

To Date: Jul 24 2006

Number of Adults:

Number of children:

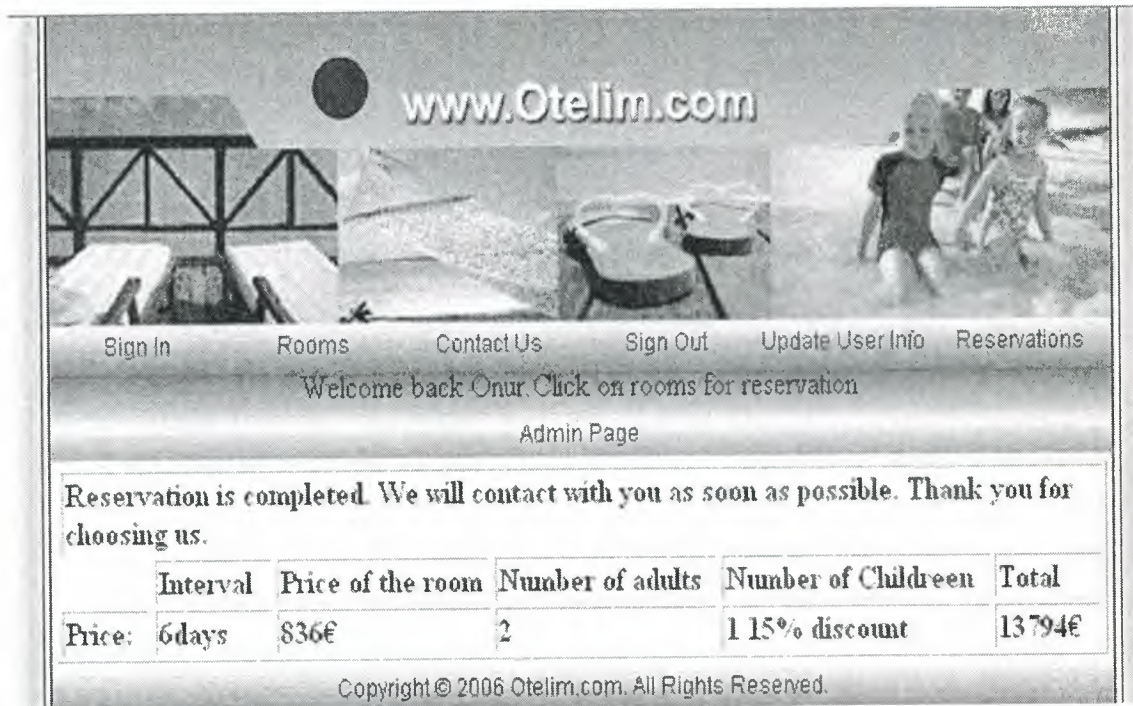
Reserve

Copyright © 2006 Otelim.com. All Rights Reserved.

Figure3.6(Reserve.asp)

3.7 Reservep.asp

If there is any empty room in the interval that the user choosed in reserve.asp the reservation is made otherwise error message comes.



The screenshot shows the website www.Otelim.com with a header featuring three images: a pool, a room, and children. A navigation bar includes links for Sign In, Rooms, Contact Us, Sign Out, Update User Info, and Reservations. A welcome message for 'Onur' is displayed, along with an 'Admin Page' link. A confirmation message states: 'Reservation is completed. We will contact with you as soon as possible. Thank you for choosing us.' Below this is a table summarizing the reservation details.


	Interval	Price of the room	Number of adults	Number of Children	Total
Price:	6days	836€	2	1 15% discount	13 794€

Copyright © 2006 Otelim.com. All Rights Reserved.

Figure 3.7(Reservep.asp)

3.8 Res.asp

This page is designed for the users who want to see the reservations made.(Figure3.8).



The screenshot shows the website **www.Otelim.com** with a navigation bar containing links: Sign In, Rooms, Contact Us, Sign Out, Update User Info, and Reservations. A welcome message reads: "Welcome back Onur. Click on rooms for reservation". Below this is an "Admin Page" section containing a table with reservation details.

	Interval	Price of the room	Number of adults	Number of Children	Total
Price:	6days	5040€	2	2 15% discount	5040€
	Interval	Price of the room	Number of adults	Number of Children	Total
Price:	6days	13794€	2	1 15% discount	13794€

Contact us to change the reservation date and for the payment options.
Copyright © 2006 Otelim.com. All Rights Reserved.

Figure 3.8(Res.asp)

3.9 Contactus.asp

The contact information is provided in this page for the users to contact with the administrators of the page.(Figure 3.9).

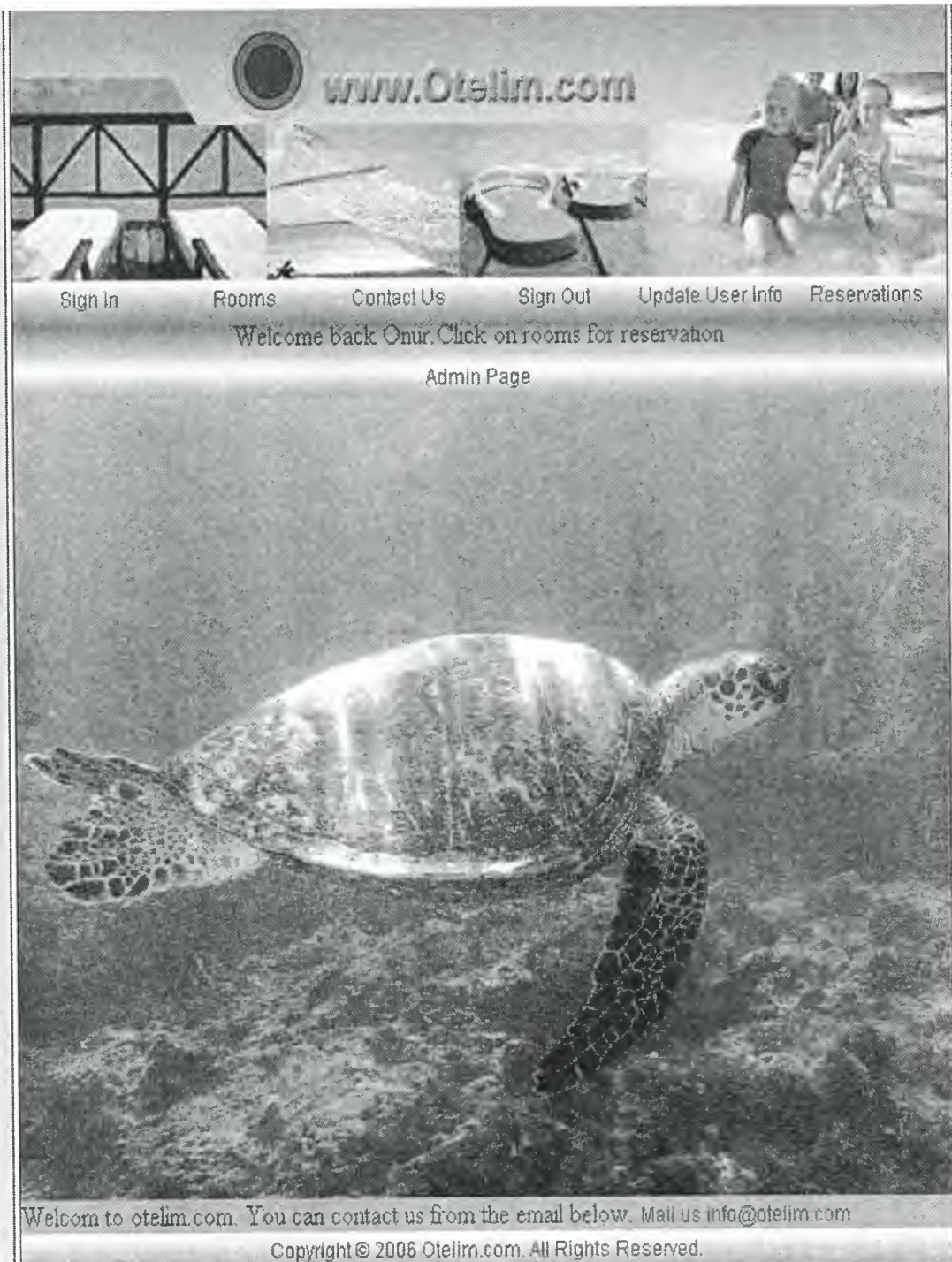


Figure 3.9 (Contactus.asp)

CHAPTER IV

4.1 THE ADMINISTRATION OF THE WEBSITE (Otelim.com)

This part of the program is for the administration of the pages. The administrator can manage the users, rooms, reservations.

In this page there are links at the top for different purposes (Figure4.1).

The links and their purposes are given below.

1. Users: This link is for the administrator to see the user accounts. The administrator can see the user accounts and can delete the unwelcome user accounts (Figure4.2).
2. Rooms: This link is for seeing the rooms and if there is something wrong, to change the room details (Figure4.3).
3. Reservations: This link take the administrator to the list of reservations. If there is a new reservation or if he wants to delete or update a reservation(Figure4.4).



Figure4.1(Administrator_of_the_site.asp)

Users			Rooms			Reservations			
UserType	Username	Password	Name	Surname	E mail	Address	Tel	Update	Delete
User	sureyya	1234	Sureyya	Gokmen	sureyyagokmen@mynet.com	Kibris	2536473290918	Update	Delete
User	onurcuk	456	Onur	Gokmen	onur.gokmen@gmail.com	as	as	Update	Delete
Admin	admin	123456	Onur	Gokmen	admin@otelim.com	Izmir	05338657020	Update	Delete
Staff	Neco	123	Necip	Soy	Necipsoy34@hotmail.com	Istanbul	0212748392	Update	Delete
Staff	ozgur	123	Ozgur	Vurgun	asd	ads	asd	Update	Delete

Figure4.2(Administrator_of_the_site.asp list of the users)

Users		Rooms		Reservations					
Room Type	Picture	Info	Long info	Price	RoomID	Stair	Update	Delete	
King Suite (max 2)	roompictures/king suit bedroom_resize[2].jpg	King Suite-1 King-sea view- balcony-sitting area -195 sq m-AC-2 TV- safe-minibar- harddryer	King Suite has 1 King bed with a sea view, balcony, and sitting area. Room is 195 square meters and has 1 and a half bathrooms with hairdryer. AC, safe, and minibar. Children under 7 stay free in the room with parents and in existing bedding. Children age 7-12 receive a 50 percent reduction.	1254	102	2	Update	Delete	
Standard Maximum Occupation 3	roompictures/standart room2_resize[3].jpg	Standard Room-2 Twins- balcony-AC- sea view in some rooms- minibar-37 sq meters-plasma tv-safe	Standard Room has 2 Twin beds or a King. Rooms are 37 meters squared. Some rooms have a sea view. All rooms have balcony, safe, air conditioning, minibar, 28 inch plasma TV, and harddryer. Children under 7 stay free in the room with parents and in existing bedding. Children age 7-12 receive a 50 percent reduction.	240	103	1	Update	Delete	

Figure4.3(Administrator_of_the_site.asp list of the rooms)



Figure 4.4 (Administrator_of_the_site.asp signout)

The screenshot shows the same Otelim.com website, but the "Admin Page" section contains a registration form. The form fields are as follows:

Username:	admin
Password:	
Re-type Password:	
Name:	Onur
Surname:	Gokmen
E-mail:	admin@otelim.com
Address:	Izmir
Tel:	05338657020
<input type="button" value="Register"/>	

Below the form, the copyright notice "Copyright © 2006 Otelim.com. All Rights Reserved." is visible.

Figure4.5 (Administrator_of_the_site.asp changing admin account)

Users		Rooms			Reservations					
RoomID	Username	Number of Adults	Number of Children	Total	Start Date	Finishdate	Reservation Date	Reservation Time	Update	Delete
105	onurcuk	1	1	14280	22/06/2006	26/07/2006	22/06/2006	19:14:43	Update	Delete
104	admin	2	2	5040	24/06/2006	30/06/2006	24/06/2006	14:09:41	Update	Delete
107	sureyya	2	1	28143.5	24/06/2006	07/08/2006	24/06/2006	14:14:13	Update	Delete
202	admin	2	1	13794	24/06/2006	30/06/2006	24/06/2006	14:29:07	Update	Delete

Figure4.6 (Administrator_of_the_site.asp list of reservations)

4.2 INCLUDED PAGES

4.2.1 Ust.asp

This page is included in each page. It consists the animation at the top the links for the users (Figure 4.2.1).



Figure4.2.1(Ust.asp)

4.2.2 Alt.asp

This page is at the bottom of each page. It consists Figure 4.2.2.



Figure 4.2.2 (Alt.asp)

4.2.3 Functions.asp

This page consists of procedures and functions that is commonly used in the pages. The pages that include this page can use these functions.

4.2.4 Style.css

The style sheets are for setting a page's page formatting. It means that if you write a specification for a tag all the tags will be so.

Ex:

```
A
{
  FONT-WEIGHT: bolder;
  COLOR: #0055ff;
  TEXT-DECORATION: none
}
```

This code makes all the Hyperlinks bolder, changes their colour to that hex code and will remove the underline that is put by the default.

This style sheet is included in the top.asp therefore all the pages include this style.

4.2.5 Rooms.asp

This page shows the room's detailed information. Change the information of the all rooms.

Users		Rooms		Reservations				
Room Type	Picture	Info	Long info	Price	RoomID	Stair	Update	Delete
King Suite (max 2)	roompictures/king suit bedroom_resize[2].jpg	King Suite-1 King-sea view- balcony-sitting area -195 sq m-AC-2 TV- safe-minibar- hairdryer	King Suite has 1 King bed with a sea view, balcony, and sitting area. Room is 195 square meters and has 1 and a half bathrooms with hairdryer, AC, safe, and minibar. Children under 7 stay free in the room with parents and in existing bedding. Children age 7-12 receive a 50 percent reduction.	1254	102	2	Update	Delete
Standard Maximum Occupation 3	roompictures/standart room2_resize[3].jpg	Standard Room-2 Twins- balcony-AC- sea view in some rooms- minibar-37 sq meters-plasma tv-safe	Standard Room has 2 Twin beds or a King. Rooms are 37 meters squared. Some rooms have a sea view. All rooms have balcony, safe, air conditioning, minibar, 28 inch plasma TV, and hairdryer. Children under 7 stay free in the room with parents and in existing bedding. Children age 7-12 receive a 50 percent reduction.	240	103	1	Update	Delete

Figure 4.2.5(Rooms.asp)

CONCLUSION

SUMMARY

In this graduation project I have studied an Hotel Reservation program– Otelim.com. I have made extensive use of the network technology more specifically, Internet technology (see previous chapters for further information) so as to carry out the business functions through ASP; that is the internet technology has been used for commercial activities.

The study conducts a set of linked themed pages which can be stored on a web server; now we can easily say that the website through internet has commercial opportunity to business; that is, many individuals use the internet for shopping; so, the project completed can offer buyers convenience. To compare prices and make purchases, without having to leave their homes or offices, the buyers can visit the website provided.

I have made the graduation project possible in the sections and chapters of the study as indicated in the following.

Logically speaking, the program I have desired and completed can easily provide both the sellers and the buyers with the facilities for shopping.

Recommendation

Through **Otelim.com** we can offer the consumers to reserve a room in the hotel from the house. I myself strongly recommend the consumers for reserve the room through the program.

REFERENCE

Abbot,Carl, et al. 2003. *Encarta Encyclopedia Deluxe 2003*. Microsoft Corporation.

<http://www.microsoft.com./inswich>

Boyle,Terence,et al.2003. *Encyclopedia Britanica 2003 Ultimate treference Suite*.

Encyclopedia Britanica.Inc.

Ince, Darrel. 2001. *dictionary of the Internet*. Oxford: Oxford University Press.

Şen,O.Nihat. 2002. *Oracle (9i)*.. İstanbul: Beta Basım Yayın Dağıtım A.Ş

Walther, Stephen. Jonathan Levine (translated by Taylan Yemliha). 2001.*ASP ile*

E-Ticaret Programcılığı.İstanbul: Sistem Yayıncılık.

Yanık,Memik.2000. *Microsoft Visual Basic 6.0*. İstanbul: Beta Basım Yayın Dağıtım A.Ş..

APENDIX 1 USER SIDE PAGE'S CODES

DEFAULT.ASP

```
<!--#include file="includes/ust.asp"-->

<tr>

<td width="30" background="images/leftshadow.jpg">

</td>

<td>

<!--#include file="includes/rooms.asp"-->

</td>

<td width="30" background="images/rightshadow.jpg">

</td>

</tr>

<!--

<tr>

<td width="30" background="images/leftshadow.jpg">

</td>

<td width="30" background="images/rightshadow.jpg">

</td>

</tr>

-->
```



```
<!--#include file="includes/alt.asp"-->
```

SIGNIN.ASP

```
<!--#include file="includes/ust.asp"-->
```

```
<tr>
```

```
<td width="30" background="images/leftshadow.jpg">
```

```
</td>
```

```
<td align="center">
```

```
<form method="post" action="signinp.asp">
```

```
<table bgcolor="#FF9900" cellpadding="1" cellspacing="1">
```

```
<tr>
```

```
<td align="left">
```

Username:

```
</td>
```

```
<td>
```

```
<input type="text" name="username">
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td align="left">
```

Password:

```

</td>

<td>

<input type="password" name="password">

</td>

</tr>

<tr>

<td colspan="2" align="center"><a href="register.asp">New User</a></td>

</tr>

<tr>

<td colspan="2" align="center"><a href="forgotpassword.asp">Forgot
Password</a></td>

</tr>

<tr>

<td colspan="2" align="center"><input type="submit" value="Sign In"></td>

</tr>

</table>

</form>

</td>

<td width="30" background="images/rightshadow.jpg">

</td>

</tr>

```

```
<!--

<tr>

<td width="30" background="images/leftshadow.jpg">

</td>

<td width="30" background="images/rightshadow.jpg">

</td>

</tr>

-->

<!--#include file="includes/alt.asp"-->
```

SIGNINP.ASP

```
<!--#include file="includes/ust.asp"-->

<%

username = request.Form("username")

password = request.Form("password")

set conn = DBConnexion()

sql = "SELECT
UserID,UserName,UserLevel,UserName,Name,Surname,Email,address,tel FROM
Users WHERE Username = " & username & " AND Password = " & password & ""

set rs = conn.execute(sql)

if rs.eof then

hata = true
```



```

else

Session("UserID") = rs("UserID")

Session("UserName") = rs("Username")

Session("Uname") = rs("Name")

Session("Email") = rs("Email")

Session("Surname") = rs("Surname")

Session("UserLevel") = rs("UserLevel")

Session("Address") = rs("address")

Session("tel") = rs("tel")

response.Redirect("default.asp")

end if

%>

<%if hata then%>

<tr>

<td width="30" background="images/leftshadow.jpg">

</td>

<td>

Wrong username and/or password.<a href="javascript:history.go(-1)">Back</a>

</td>

<td width="30" background="images/rightshadow.jpg">

```

</td>

</tr>

<%end if%>

<!--#include file="includes/alt.asp"-->

REGISTER.ASP

<!--#include file="includes/ust.asp"-->

<tr>

<td width="30" background="images/leftshadow.jpg">

</td>

<td align="center">

<form method="post" action="registerp.asp">

<table bgcolor="#FF9900">

<tr>

<td align="left">

Username:

</td>

<td>

<input type="text" name="username">

</td>

</tr>

<tr>

<td align="left">

Password:

</td>

<td>

<input type="password" name="password">

</td>

</tr>

<tr>

<td align="left">

Re-type Password:

</td>

<td>

<input type="password" name="password1">

</td>

</tr>

<tr>

<td align="left">

Name:

</td>

<td>

<input type="text" name="uname">

</td>

</tr>

<tr>

<td align="left">

Surname:

</td>

<td>

<input type="text" name="surname">

</td>

</tr>

<tr>

<td align="left">

E-mail:

</td>

<td>

<input type="text" name="email">

</td>

</tr>

```
<tr>
```

```
<td align="left">
```

Address:

```
</td>
```

```
<td>
```

```
<input type="text" name="address">
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td align="left">
```

Tel:

```
</td>
```

```
<td>
```

```
<input type="text" name="tel">
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td colspan="2">
```

```
<input type="submit" value="Register">
```

```
</td>
```

```

</tr>

</table>

</form>

</td>

<td width="30" background="images/rightshadow.jpg">

</td>

</tr>

<!--

<tr>

<td width="30" background="images/leftshadow.jpg">

</td>

<td width="30" background="images/rightshadow.jpg">

</td>

</tr>

-->

<!--#include file="includes/alt.asp"-->

```

REGISTERP.ASP

```

<!--#include file="includes/ust.asp"-->
<%
username = request.Form("username")
uname = request.Form("uname")
Surname = request.Form("Surname")

```




```

Email = request.Form("Email")
Password = request.Form("Password")
Password1 = request.Form("Password1")
address = request.Form("address")
tel = request.Form("tel")
hata = false
e = false
r = false
set conn = dbconnexion()
sqle = "SELECT Email FROM Users WHERE Email = " & Email & ""
set rs = conn.execute(sqle)
if not rs.eof then
e = true
end if
if username = "" or uname = "" or Surname = "" or Email = "" or Password = "" or
Password1 = "" or Password <> Password1 then
r = true
end if
sql = "INSERT INTO
Users(username,userlevel,password,name,surname,email,address,tel) VALUES (" &_
"" & username & "," & 1 & "," & password & "," & uname & "," & Surname
& "," & Email & "," & address & "," & tel & ")"
if not e and not r then
conn.execute(sql)
end if
hata = false
if err.number <> 0 then
hata = true
end if
conn.close
set conn = nothing
%>
<tr>
<td width="30" background="images/leftshadow.jpg">

```

```

</td>
<td valign="top" class="hata" align="center">
<%if hata then%>
Sorry there was a problem try again later.<a href="javascript:history.go(-1)">Back</a>
<%response.Write(err.description)%>
<%elseif e then%>
The email exists change your email.<a href="javascript:history.go(-1)">Back</a>
<%elseif r then%>
One of the required field is empty.<a href="javascript:history.go(-1)">Back</a>
<%else%>
Registration finished succesfully.
<table width="100" border="0">
<tr>
<td>Username:</td><td><%=username%></td>
</tr>
<tr>
<td>Password:</td><td><%=password%></td>
</tr>
<tr>
<td>Name:</td><td><%=uname%></td>
</tr>
<tr>
<td>Surname:</td><td><%=surname%></td>
</tr>
<tr>
<td>E mail:</td><td><%=email%></td>
</tr>
<tr>
<td>Address:</td><td><%=address%></td>
</tr>
<tr>
<td>Tel:</td><td><%=tel%></td>
</tr>
</table>

```

```
<%end if%>
</td>
<td width="30" background="images/rightshadow.jpg">
</td>
</tr>
<!--#include file="includes/alt.asp"-->
```

SIGNOUT.ASP

```
<!--#include file="includes/ust.asp"-->

<%

Session.Contents.removeall

%>

<tr>

<td width="30" background="images/leftshadow.jpg">

</td>

<td>

You are succesfully logged out.

</td>

<td width="30" background="images/rightshadow.jpg">

</td>

</tr>

<!--#include file="includes/alt.asp"-->
```


FORGOTPASSWORD.ASP

```
<!--#include file="includes/ust.asp"-->

<tr>

<td width="30" background="images/leftshadow.jpg">

</td>

<td align="center" class="tabloicerik">

<form method="post" action="forgotpasswordp.asp">

<table>

<tr>

<td>

Email:

</td>

<td>

<input type="text" name="email" maxlength="50">

</td>

</tr>

<tr>

<td colspan="2" align="center">

<input type="submit" value="Send">

</td>
```

```
</tr>
```

```
</table>
```

```
</form>
```

```
</td>
```

```
<td width="30" background="images/rightshadow.jpg">
```

```
</td>
```

```
</tr>
```

```
<!--#include file="includes/alt.asp"-->
```

FORGOTPASSWORDP.ASP

```
<!--#include file="includes/ust.asp"-->
```

```
<%
```

```
email = request.Form("email")
```

```
set conn = dbconnexion()
```

```
sql = "SELECT email,password FROM Users WHERE Email = '" & email & "'"
```

```
set rs = conn.execute(sql)
```

```
e = false
```

```
response.Write(rs("email"))
```

```
if rs.eof then
```

```
e = true
```

```
end if
```

if not e then

Set myMail=CreateObject("CDO.Message")

myMail.Subject="Your password"

myMail.From="onur.gokmen@gmail.com"

myMail.To=cstr(trim(rs("email")))

myMail.HTMLBody="This is an automatically generated message

Your password is :" & cstr(rs("Password"))

myMail.Send

set myMail=nothing

end if

%>

<tr>

<td width="30" background="images/leftshadow.jpg">

</td>

<td valign="top" class="tabloicerik">

<%if not e then%>

Mail has been sent.

<%else%>

Error. The mail address that you entered is not in our database.

<%end if%>

</td>

<td width="30" background="images/rightshadow.jpg">

</td>

</tr>

<!--#include file="includes/alt.asp"-->

CONTACTUS.ASP

<!--#include file="includes/ust.asp"-->

<tr bgcolor="#999999">

<td width="30" background="images/leftshadow.jpg">

</td>

<td>
Welcom
to otelim.com. You can contact us from the email below.

Mail us info@otelim.com

</td>

<td width="30" background="images/rightshadow.jpg">

</td>

</tr>

<!--

<tr>

<td width="30" background="images/leftshadow.jpg">

</td>

```
<td width="30" background="images/rightshadow.jpg">
```

```
</td>
```

```
</tr>
```

```
-->
```

```
<!--#include file="includes/alt.asp"-->
```

UPDATEINFO.ASP

```
<!--#include file="includes/ust.asp"-->
```

```
<%
```

```
if Session("UserID")<>"" then
```

```
%>
```

```
<tr>
```

```
<td width="30" background="images/leftshadow.jpg">
```

```
</td>
```

```
<td align="center">
```

```
<form method="post" action="updateinfop.asp">
```

```
<table bgcolor="#FF9900">
```

```
<tr>
```

```
<td align="left">
```

```
Username:
```

```
</td>
```

<td>

<input type="text" name="username" value="<%=Session("UserName")%>">

</td>

</tr>

<tr>

<td align="left">

Password:

</td>

<td>

<input type="password" name="password">

</td>

</tr>

<tr>

<td align="left">

Re-type Password:

</td>

<td>

<input type="password" name="password1">

</td>

</tr>

<tr>

<td align="left">

Name:

</td>

<td>

<input type="text" name="uname" value="<%=Session("Uname")%>">

</td>

</tr>

<tr>

<td align="left">

Surname:

</td>

<td>

<input type="text" name="surname" value="<%=Session("Surname")%>">

</td>

</tr>

<tr>

<td align="left">

E-mail:

</td>

<td>

<input type="text" name="email" value="<%=Session("email")%>">

</td>

</tr>

<tr>

<td align="left">

Address:

</td>

<td>

<input type="text" name="address" value="<%=Session("Address")%>">

</td>

</tr>

<tr>

<td align="left">

Tel:

</td>

<td>

<input type="text" name="tel" value="<%=Session("Tel")%>">

</td>

</tr>

```

<tr>

<td colspan="2">

<input type="submit" value="Register">

</td>

</tr>

</table>

</form>

</td>

<td width="30" background="images/rightshadow.jpg">

</td>

</tr>

<!--

<tr>

<td width="30" background="images/leftshadow.jpg">

</td>

<td width="30" background="images/rightshadow.jpg">

</td>

</tr>

-->

<%else%>

```

```
<tr>
```

```
<td width="30" background="images/leftshadow.jpg">
```

```
</td>
```

```
<td align="center">
```

You can not change your information without signing in.

```
</td>
```

```
<td width="30" background="images/rightshadow.jpg">
```

```
</td>
```

```
</tr>
```

```
<%end if%>
```

```
<!--#include file="includes/alt.asp"-->
```

UPDATEINFOP.ASP

```
<!--#include file="includes/ust.asp"-->
```

```
<%
```

```
username = request.Form("username")
```

```
uname = request.Form("uname")
```

```
Surname = request.Form("Surname")
```

```
Email = request.Form("Email")
```

```
Password = request.Form("Password")
```

```
Password1 = request.Form("Password1")
```

```
address = request.Form("address")
```

```
tel = request.Form("tel")
```

```
hata = false
```

```
u = false
```

```
e = false
```



```

p = false
set conn = dbconnexion()
sqle = "SELECT Email FROM Users WHERE Email = " & Email & ""
set rs = conn.execute(sqle)
if not rs.eof and session("email") <> email then e = true
if password <> password1 then p = true
if Session("UserID") = "" or Session("UserID") = 0 then u = true
response.Write(e & " " & u & " " & p)
sql = "UPDATE Users SET UserName = " & username & ",Name=" & uname &
",surname=" & surname & ",email=" & email & ",Password = " & Password &
",address=" & address & ",tel=" & tel & " WHERE UserID=" & Session("UserID")
'response.Write(sql)
'response.End()
if not u and not e and not p then
conn.execute(sql)
Session("UserName") = username
Session("Uname") = uname
Session("Email") = email
Session("Surname") = surname
Session("Address") = address
Session("Tel") = tel
end if
hata = false
if err.number <> 0 then
hata = true
end if
conn.close
set conn = nothing
%>
<tr>
<td width="30" background="images/leftshadow.jpg">
</td>
<td align="center">
<%if u then%>Change Username<%end if%>

```

```

<%if e then%>Email exists<%end if%>
<%if p then%>Passwords are not same<%end if%>
<%if hata then%>Error occured try again later<%end if%><br>
<%if e or p or u then%><a href="javascript:history.go(-1)">Back</a><%end if%>
<%if not e and not u and not p and not hata then%>Update is made succesfully<%end
if%>
</td>
<td width="30" background="images/rightshadow.jpg">
</td>
</tr>
<!--#include file="includes/alt.asp"-->

```

RES.ASP

```

<!--#include file="includes/ust.asp"-->

<%

set conn = dbconnexion()

sql      =      "SELECT      *      FROM      Users,Reservations      WHERE
Reservations.UserID=Users.UserID AND Users.UserID=" & Session("UserID")

set rs = conn.execute(sql)

'response.Write(sql)

'response.End()

sure = datediff("d",rs("StartDate"),rs("Finishdate"))

%>

<tr>

<td width="30" background="images/leftshadow.jpg">

</td>

```

<td>

<table border="1">

<%while not rs.eof%>

<tr>

<td>

</td>

<td>Interval</td>

<td>Price of the room

</td>

<td>Number of adults

</td>

<td>Number of Children

</td>

<td>Total

</td>

</tr>

<tr>

<td>Price:</td> <td><%=sure%>days

</td>

<td><%=rs("total")%>€

</td>

<td><%=rs("noofadults")%>

</td>

td><%=rs("noofchildren")%> 15% discount

</td>

<td><%=rs("total")%>€

</td>

</tr>

<%

rs.movenext

wend

%>

</table>

Contact us to change the reservation date and for the payment options.

</td>

<td width="30" background="images/rightshadow.jpg">

</td>

</tr>

<!--#include file="includes/alt.asp"-->

RESERVE.ASP

```
<!--#include file="includes/ust.asp"-->
```

```
<tr>
```

```
<td width="30" background="images/leftshadow.jpg">
```

```
</td>
```

```
<td>
```

```
<%if Session("UserID")<>" " then%>
```

```
<form method="post" action="reservep.asp?id=<%=request.QueryString("id")%>">
```

```
<table align="center">
```

```
<tr>
```

```
<td>From Date:
```

```
</td>
```

```
<td><scriptlanguage="javascript">DateInput('FromDate',true'DD/MM/YYYY','<%=date%>')</script>
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td>To Date:
```

```
</td>
```

```
<td><scriptlanguage="javascript">DateInput('ToDate',true,'DD/MM/YYYY','<%=date+30%>')
```

</script>

</td>

</tr>

<tr>

<td>Number of Adults:

</td>

<td>

<input type="text" name="noofadults">

</td>

</tr>

<tr>

<td>Number of children:

</td>

<td>

<input type="text" name="noofchildren">

</td>

</tr>

<tr>

<td colspan="2" align="center"><input type="submit" value=" Reserve ">

</td>

</tr>

</table>

</form>

<%else%>

You are not signed in please click here.

<%end if%>

</td>

<td width="30" background="images/rightshadow.jpg">

</td>

</tr>

<!--#include file="includes/alt.asp"-->

RESERVEP.ASP

<!--#include file="includes/ust.asp"-->

<%

FromDate = cdate(request.Form("FromDate"))

ToDate = cdate(request.Form("ToDate"))

RoomTypeID = Request.QueryString("ID")

NoofAdults = cint(Request.Form("NoofAdults"))

NoofChildreen = cint(Request.Form("NoofChildreen"))

ID = request.QueryString("ID")

```

set conn = dbconnexion()

sql = "SELECT * FROM Rooms WHERE RoomID NOT IN(SELECT Rooms.RoomID
FROM                      Reservations,RoomTypes,Rooms                WHERE
Rooms.RoomTypeID=RoomTypes.RoomTypeID                                AND
Rooms.RoomID=Reservations.RoomID  AND #" & sqldate(FromDate) & "#
BETWEEN Reservations.StartDate AND Reservations.FinishDate AND #" &
sqldate(ToDate) & "# BETWEEN Reservations.StartDate AND
Reservations.FinishDate)AND Rooms.RoomTypeID=" & RoomTypeID & ""

set rs = conn.execute(sql)

surehatasi = false

sayihatasi = false

sure = datediff("d",FromDate,ToDate)

'response.Write(sure & "<br>" & fromdate & "<br>" & todate & "<br>")

'response.Write(sql)

if rs.eof then

surehatasi = true

end if

'response.Write(sure)

if sure < 3 then

surehatasi = true

end if

if noofadults<1 then

sayihatasi = true

```



```

end if

if noofchildren<1 then

sayihatasi = true

end if

sqlroom = "SELECT Price FROM Roomtypes WHERE RoomTypeID=" & ID

set rsroom = conn.execute(sqlroom)

'response.Write(sql)

'response.End()

'response.Write(rs("Price"))

%>

<tr>

<td width="30" background="images/leftshadow.jpg">

</td>

<td>

<table align="center">

<tr>

<td>

<b>

<%if sayihatasi then%>

<font color="#FF0000">Number of adults and/or children is not valid.

```

```

<a                                href="javascript:history.go(-1)"
onMouseOver="window.status='DriverInfo.asp';return true">Back

</a>

</font>

<%end if%>

<%if surehatasi then%>

<font color="#FF0000">Date interval is not valid.

<a                                href="javascript:history.go(-1)"
onMouseOver="window.status='DriverInfo.asp';return true">Back

</a>

</font>

<%end if%>

<%if surehatasi=false and sayihatasi=false then%>

<%

total                                =

(sure*rsroom("Price")*noofadults)+(sure*rsroom("Price")*(noofchildren*0.75))

sqlreserve                        =                "INSERT                INTO
Reservations(RoomID,UserID,NoofAdults,NoofChildren,Total,StartDate,FinishDate,R
eservationDate,Reservationtime)  VALUES  ("  &  rs("RoomID")  &  ","  &
Session("UserId")  &  ","  &  noofadults  &  ","  &  noofchildren  &  ","  &  Total  &  ","  &
FromDate  &  "#,#"  &  ToDate  &  "#,#"  &  sqldate(date)  &  "#,#"  &  time  &  "#)"

conn.execute(sqlreserve)

%>

```

```
<table border="1">
```

```
<tr>
```

```
<td colspan="6">
```

```
<b>Reservation is completed. We will contact with you as soon as possible. Thank you  
for choosing us.
```

```
</b>
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td>
```

```
</td>
```

```
<td>Interval
```

```
</td>
```

```
<td>Price of the room
```

```
</td>
```

```
<td>Number of adults
```

```
</td>
```

```
<td>Number of Children
```

```
</td>
```

```
<td>Total
```

```
</td>
```

</tr>

<tr>

<td>Price:

</td> <td>

<%=sure%>days</td><td><%=rsroom("Price")%>€

</td>

<td><%=noofadults%>

</td>

<td><%=noofchildren%> 15%discount

</td>

<td><%=(sure*rsroom("Price")*noofadults)+(sure*rsroom("Price")*(noofchildren*0.7
5))%>€

</td>

</tr>

</table>

<%end if%>

</td>

</tr>

</table>

</td>

<td width="30" background="images/rightshadow.jpg">

</td>

</tr>

<%

conn.close

set conn = nothing

%>

<!--#include file="includes/alt.asp"-->

APENDIX 2 INCLUDED PAGES

UST.ASP

```
<%@LANGUAGE="VBSCRIPT"%>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<html>
```

```
<head>
```

```
<title>Otelim.com</title>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
```

```
<link href="includes/style.css" rel="stylesheet" type="text/css">
```

```
<script type="text/javascript" src="includes/calendarDateInput.js"></script>
```

```
</head>
```

```
<!--#include file="functions.asp"-->
```

```
<body topmargin="0" leftmargin="0" rightmargin="0" bottommargin="0"
bgcolor="#FFFFFF">
```

```
<table class="anatabloust" align="center" cellpadding="0" cellspacing="0"
width="660px">
```

```
<tr>
```

```
<td width="30px" background="images/leftshadow.jpg">&nbsp;
```

```
</td>
```

```
<td>
```

```
<object          classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#ve
rsion=6,0,29,0" width="600" height="154">
```

```
<param name="movie" value="animation/banner.swf">
```

```
<param name="quality" value="high">
```

```
<embed          src="animation/banner.swf"          quality="high"
pluginspage="http://www.macromedia.com/go/getflashplayer"  type="application/x-
shockwave-flash" width="600" height="154"></embed>
```

```
</object>
```

```
</td>
```

```
<td width="30px" background="images/rightshadow.jpg">
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td width="30" background="images/leftshadow.jpg">
```

```
</td>
```

```
<td width="600px">
```

```
<table width="100%" border="0" background="images/bannerbar.jpg">
```

```
<tr>
```

```
<td width="16%" align="center">
```

```
<a href="signin.asp">Sign In
```

```
</a>
```

</td>

<td width="16%" align="center">

Rooms

</td>

<td width="16%" align="center">

Contact Us

</td>

<td width="16%" align="center">

Sign Out

</td>

<td width="16%" align="center">

Update User Info

</td>

<td width="16%" align="center">

Reservations


```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td colspan="6" align="center">
```

```
<%if Session("UserID")="" then%>
```

Welcome to www.otelim.com If you are a registered user please click Sign in to login.

```
<%else%>
```

Welcome back <%=Session("uname")%>.Click on rooms for reservation

```
<%end if%>
```

```
</td>
```

```
</tr>
```

```
<%if Session("UserLevel")>1 then%>
```

```
<tr>
```

```
<td colspan="6" align="center">
```

```
<a href="admin.asp">Admin Page
```

```
</a>
```

```
</td>
```

```
</tr>
```

```
<%end if%>
```

```
</table>
```

</td>

<td width="30" background="images/rightshadow.jpg">

</td>

</tr>

ALT.ASP

<tr height="20">

<td width="30" background="images/leftshadow.jpg">

</td>

<td align="center" class="icerik" width="600">

<table width="100%" border="0" background="images/bannerbar.jpg">

<tr>

<td>

Copyright © 2006 Otelim.com. All Rights Reserved.

</td>

</tr>

</table>

</td>

<td width="30" background="images/rightshadow.jpg">

</td>

</tr>

</table>

</body>

</html>

ROOMS.ASP

<%

pg = request.QueryString("pg")

set conn = dbconnexion()

sql = "SELECT * FROM ROOMTYPES ORDER BY RoomTypeID"

Set rs = Server.CreateObject("ADODB.Recordset")

rs.ActiveConnection = Conn

rs.CursorType = adOpenStatic

rs.PageSize = 20

rs.open sql

rs.AbsolutePage = pg

%>

<table width="100%">

<tr>

<td align="center">Room Types

</td>

</tr>

<tr>

<td>

<table width="100%" height="119">

<%

while not rs.eof and rowCount < rs.pagesize

rowcount = rowcount + 1

%>

<tr>

<td width="150" rowspan="3">" width="150" height="100">

</td>

```

<td width="305" height="24" valign="top"><%=rs("roomtype")%>
</td>
</tr>
<tr>
<td valign="top">
<table>
<tr>
<td width="100"><%=rs("Price")%>$/day</td>
<td>
<a href="reserve.asp?id=<%=rs("roomtypeID")%>"></a>
</td>
</tr>
</table>
</td>
</tr>
<tr>
<td valign="top"><%=rs("Info")%></td>
</tr>
<tr>
<td colspan="2"><%=rs("longinfo")%></td>
</tr>
<%
rs.movenext
wend
%>
<%
IF rs.PageCount > 1 THEN
FOR i = 1 to rs.PageCount
IF i <> cINT( pg ) THEN
%>
<a href="default.asp?pg=<%=i%>"><%=i%></a>&nbsp;
<%
ELSE

```



```
%>
<b><%=i%></b>&nbsp;
<%
END IF
NEXT
END IF
%>
</table>
</td>
</tr>
</table>
<%
conn.close
set conn = nothing
%>
```

FUNCTIONS.ASP

```
<!--#include file="ADOVBS.INC"-->

<%

on error resume next

SLC

'-----

' steStripHTML

'      Strip out all HTML from the content parameter (passed by ref)

Function SetBool(Byval bdeger)

if bdeger = "" or bdeger = null then

setbool = "False"
```

else

setbool = "True"

end if

End Function

Function sqlDate(Byval tarih)

sqlDate = month(tarih) & "/" & day(tarih) & "/" & year(tarih)

End Function

Sub LoginCheck()

if Session("kullan") = "" then

response.Write("You are not allowed to enter this part of the website.Login")

response.End()

end if

End Sub

Sub isSuper()

if Session("super") <> true then

response.Write("You are not allowed to see this part of the website")

response.End()

end if

End Sub

Sub checkValue(deger)

if deger = true then

response.Write("checked")

end if

End Sub

Function fixQuotes(theString)

fixQuotes=steStripHTML(trim(REPLACE(theString, "\"", "")))

End Function

Function SetNum(Byval bdeger)

if bdeger = "" then

SetNum = "0"

else

SetNum = bdeger

end if

End Function

Function SetdbNum(Byval bdeger)

if bdeger = "" or bdeger = null then

SetNum = 0

else

SetNum = bdeger

end if

End Function

Function SLC()

Session.LCID = 2057

Session.Timeout = 60

End Function

Function DBConnexion()

Dim conn

Set conn = Server.CreateObject("ADODB.Connection")

sconnstring="DRIVER={Microsoft Access Driver (*.mdb)}; " & _

"DBQ=" & Server.MapPath("db/reservation.mdb")

On Error Resume Next

conn.open sconnstring

If Err.number <> 0 Then

Response.Write "
" & GetTranslation("LANG_SQL_ERROR")

If IsAuthorized(ROLE_ADMIN) Then

Response.Write " : " & Err.Description & "
"

Response.Write "Connexion String : " & sConnString & "
"

End If

Err.Clear

End If

On Error Goto 0

set DBConnexion = conn

End Function

%>

STLYE.CSS

.anatabloust

{

letter-spacing: normal;

vertical-align: top;

word-spacing: normal

}

.anatabloorta

{

letter-spacing: normal;

vertical-align: top;

word-spacing: normal;

}

.anatabloalt

{

letter-spacing: normal;

vertical-align: top;

word-spacing: normal;

}

td.icerik

{

font-family: Arial, Helvetica, sans-serif;

font-size: 12px;

}

tr.forms

{

font-family: Arial, Helvetica, sans-serif;

font-size: 12px;

color: #FFFFFF;

background-color: #003366;

}

tr.uyari

{

font-family: Arial, Helvetica, sans-serif;

font-size: 12px;

color: #CCCCCC;

```
background-color:#000033;
```

```
}
```

```
tr.top
```

```
{
```

```
background-color:#000033;
```

```
}
```

```
tr.fiyat
```

```
{
```

```
font-family:Arial, Helvetica, sans-serif;
```

```
font-size:12px;
```

```
color:#FFFFFF;
```

```
background-color:#666666;
```

```
}
```

```
tr.list
```

```
{
```

```
font-family:Arial, Helvetica, sans-serif;
```

```
font-size:12px;
```

```
color:#FFFFFF;
```

```
background-color:#333333;
```

```
}
```

td.error

{

font-family: Arial, Helvetica, sans-serif;

font-size: 14px;

color: #FF0000;

font-weight: bold;

background-color: #FFFFFF;

}

a.top

{

font-family: Arial, Helvetica, sans-serif;

font-size: 12px;

color: #999999;

text-decoration: none;

}

a.top:hover

{

font-size: 12px;

color: #000066;

text-decoration: underline;

}

a

{

font-family: Arial, Helvetica, sans-serif;

font-size: 12px;

color: #333333;

text-decoration: none;

}

a:hover

{

font-size: 12px;

color: #FFFFFF;

text-decoration: underline;

}

APENDIX 3 THE CODES FOR THE ADMINISTRATION PART

ADMINISTRATION PART

ADMINTOP.ASP

```
<%@LANGUAGE="VBSCRIPT"%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<title>Otelim.com Administration</title>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">

<link href="includes/style.css" rel="stylesheet" type="text/css">

<script type="text/javascript" src="includes/calendarDateInput.js"></script>

</head>

<!--#include file="functions.asp"-->

<%

if Session("UserLevel")>1 then

%>

<table border="1" cellpadding="0" cellspacing="0" align="center" width="100%"
bgcolor="#999999">

<tr>
```

```
<td width="33%"><a href="users.asp">Users</a></td><td width="33%"><a href="rooms.asp">Rooms</a></td><td width="33%"><a href="Reservations.asp">Reservations</a></td>
```

```
</tr>
```

```
</table>
```

```
<%
```

```
else
```

```
response.End()
```

```
end if
```

```
%>
```

USERS.ASP

```
<!--#include file="includes/admintop.asp"-->
```

```
<%
```

```
set conn = dbconnexion()
```

```
sql = "SELECT * FROM Users,UserLevels WHERE  
Users.UserLevel=UserLevels.UserLevelID"
```

```
set rs = conn.execute(sql)
```

```
%>
```

```
<table border="1" cellpadding="0" cellspacing="0" width="100%"  
bgcolor="#FF9900">
```

```
<tr>
```

```
<td>UserType</td><td>Username</td><td>Password</td><td>Name</td><td>Surname</td><td>Email</td><td>Address</td><td>Tel</td><td>Update</td><td>Delete</td>
```

```
</tr>
```

```
<%
```

```
while not rs.eof
```

```
%>
```

```
<tr>
```

```
<td><%=rs("UserLevelName")%>
```

```
</td>
```

```
<td>
```

```
<%=rs("Username")%>
```

```
</td>
```

```
<td>
```

```
<%=rs("Password")%>
```

```
</td>
```

```
<td>
```

```
<%=rs("Name")%>
```

```
</td>
```

```
<td>
```

```
<%=rs("Surname")%>
```



```

</td>

<td>

<%=rs("Email")%>

</td>

<td>

<%=rs("Address")%>

</td>

<td>

<%=rs("Tel")%>

</td>

<td>

<%=if Session("UserLevel")>2 then%>

<a href="useru.asp?id=

<%=rs("UserID")%>

">Update</a>

</td>

<td>

<a href="userd.asp?id=<%=rs("UserID")%>

">Delete</a>

</td>

```

```
<%end if%>
```

```
</tr>
```

```
<%
```

```
rs.movenext
```

```
wend
```

```
%>
```

```
</table>
```

```
<%
```

```
conn.close
```

```
set conn = nothing
```

```
%>
```

USERD.ASP

```
<!--#include file="includes/admintop.asp"-->
```

```
<%
```

```
ID = request.QueryString("ID")
```

```
set conn = dbconnexion()
```

```
sql = "DELETE FROM Users WHERE UserID=" & ID
```

```
conn.execute(sql)
```

```
%>
```

```
User is deleted.<a href="users.asp">Back</a>
```

```

<%

conn.close

set conn = nothing

%>

USERU.ASP

<!--#include file="includes/admintop.asp"-->

<%

if Session("UserLevel")<2 then response.End()

ID = request.QueryString("ID")

set conn = dbconnexion()

sql = "SELECT * FROM Users WHERE UserID=" & ID

set rs = conn.execute(sql)

%>

<form method="post" action="userup.asp?ID=<%=ID%>">

<table bgcolor="#FF9900" align="center">

<tr>

<td align="left">

Username:

</td>

<td>

```

```
<input type="text" name="username" value="<%=rs("UserName")%>">
```

```
</td>
```

```
</tr>
```

```
<%
```

```
sql = "SELECT * FROM UserLevels"
```

```
set rslevel = conn.execute(sql)
```

```
%>
```

```
<tr>
```

```
<td>
```

```
User Level:
```

```
</td>
```

```
<td>
```

```
<select name="UserLevel">
```

```
<%=while not rslevel.eof%>
```

```
<option value="<%=rslevel("UserLevelID")%>">
```

```
<%=rslevel("UserLevelName")%>
```

```
</option>
```

```
<%
```

```
rslevel.movenext
```

```
wend
```

%>

</select>

</td>

</tr>

<tr>

<td align="left">

Password:

</td>

<td>

<input type="text" name="password" value="<%=rs("password")%>">

</td>

</tr>

<tr>

<td align="left">

Name:

</td>

<td>

<input type="text" name="uname" value="<%=rs("Name")%>">

</td>

</tr>

<tr>

<td align="left">

Surname:

</td>

<td>

<input type="text" name="surname" value="<%=rs("Surname")%>">

</td>

</tr>

<tr>

<td align="left">

E-mail:

</td>

<td>

<input type="text" name="email" value="<%=rs("email")%>">

</td>

</tr>

<tr>

<td align="left">

Address:

</td>

<td>

<input type="text" name="address" value="<%=rs("Address")%>">

</td>

</tr>

<tr>

<td align="left">

Tel:

</td>

<td>

<input type="text" name="tel" value="<%=rs("Tel")%>">

</td>

</tr>

<tr>

<td colspan="2">

<input type="submit" value="Update">

</td>

</tr>

</table>

</form>

<%=

```
conn.close
```

```
set conn = nothing
```

```
%>
```

USERUP.ASP

```
<!--#include file="includes/admintop.asp"-->
```

```
<%
```

```
set conn = dbconnexion()
```

```
ID = request.QueryString("ID")
```

```
username = request.Form("Username")
```

```
password = request.Form("Password")
```

```
userlevelID = request.Form("UserLevel")
```

```
uname = request.Form("uname")
```

```
surname = request.Form("surname")
```

```
email = request.Form("email")
```

```
address = request.Form("address")
```

```
tel = request.Form("tel")
```

```
sql = "UPDATE Users SET UserName = " & username & ",UserLevel=" & userlevelID  
& ",Name=" & uname & ",surname=" & surname & ",email=" & email & ",Password  
= " & Password & ",address=" & address & ",tel=" & tel & " WHERE UserID=" &  
ID
```

```
'response.Write(sql)
```

```
conn.execute(sql)
```

```
%>
```

```
User is updated
```

```
<a href="users.asp">Back</a>
```

```
<%
```

```
conn.close
```

```
set conn = nothing
```

```
%>
```

RESERVATIONS.ASP

```
<!--#include file="includes/admintop.asp"-->
```

```
<%
```

```
set conn = dbconnexion()
```

```
sql = "SELECT * FROM Reservations,Rooms,Users WHERE  
Reservations.RoomID=Rooms.RoomID AND Reservations.UserID=Users.UserID"
```

```
'response.Write(sql)
```

```
'response.End()
```

```
set rs = conn.execute(sql)
```

```
%>
```

```
<table border="1" cellpadding="0" cellspacing="0" width="100%"  
bgcolor="#FF9900">
```

```
<tr>
```

<td>RoomID

</td>

<td>Username

</td>

<td>Number of Adults

</td>

<td>Number of Children

</td>

<td>Total

</td>

<td>Start Date

</td>

<td>Finishdate

</td>

<td>Reservation Date

</td>

<td>Reservation Time

</td>

<td>Update

</td>

<td>Delete

</td>

</tr>

<%

while not rs.eof

%>

<tr>

<td><%=rs("RoomID")%>

</td>

<td><%=rs("Username")%>

</td>

<td><%=rs("NoofAdults")%>

</td>

<td><%=rs("NoofCHILDREEN")%>

</td>

<td><%=rs("Total")%>

</td>

<td><%=rs("StartDate")%>

</td>

<td><%=rs("FinishDate")%>

</td>

<td><%=rs("ReservationDate")%>

</td>

<td><%=rs("ReservationTime")%>

</td>

<td><%if Session("UserLevel")>2 then%>

<a

href="reservationu.asp?id=<%=rs("ReservationID")%>&rtID=<%=rs("RoomTypeID")%>">Update

</td>

<td>

<a href="reservationd.asp?id=<%=rs("ReservationID")%>">Delete

</td><%end if%>

</tr>

<%

rs.movenext

wend

%>

</table>

```
<%
```

```
conn.close
```

```
set conn = nothing
```

```
%>
```

RESERVATIOND.ASP

```
<!--#include file="includes/admintop.asp"-->
```

```
<%
```

```
ID = request.QueryString("ID")
```

```
set conn = dbconnexion()
```

```
sql = "DELETE FROM Reservations WHERE ReservationID=" & ID
```

```
conn.execute(sql)
```

```
%>
```

```
Reservations is deleted.<a href="reservations.asp">Back</a>
```

```
<%
```

```
conn.close
```

```
set conn = nothing
```

```
%>
```

RESERVATIONU.ASP

```
<!--#include file="includes/admintop.asp"-->
```

```
<%
```

```
id = request.QueryString("id")
```

```
rtid=request.QueryString("rtid")
```

```
set conn = dbconnexion()
```

```
sql = "SELECT * FROM Reservations WHERE ReservationID=" & id
```

```
set rs = conn.execute(sql)
```

```
%>
```

```
<form method="post" action="reservationup.asp?id=<%=id%>&rtid=<%=rtid%>">
```

```
<table align="center" bgcolor="#FF9900" border="1" cellpadding="0"
cellspacing="0">
```

```
<tr>
```

```
<td>From Date:</td>
```

```
<td><scriptlanguage="javascript">DateInput('FromDate',true,'DD/MM/YYYY','<%=rs(
"StartDate")%>')</script>
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td>To Date:</td>
```

```
<td><scriptlanguage="javascript">DateInput('ToDate',true,'DD/MM/YYYY','<%=rs("F
inishDate")%>')</script>
```

```
</td>
```

```
</tr>
```

```

<tr>

<td>Number of Adults:</td>

<td><input type="text" name="noofadults" value="<%=rs("NoofAdults")%>">

</td>

</tr>

<tr>

<td>Number of children:</td>

<td><input type="text" name="noofchildren" value="<%=rs("NoofChildren")%>">

</td>

</tr>

<tr>

<td colspan="2" align="center"><input type="submit" value="      Reserve      "></td>

</tr>

</table>

</form>

```

RESERVATIONUP.ASP

```

<!--#include file="includes/admintop.asp"-->

<%

FromDate = sqldate(cdate(request.Form("FromDate")))

ToDate = sqldate(cdate(request.Form("ToDate")))

```



```

RoomTypeID = Request.QueryString("ID")

NoofAdults = cint(Request.Form("NoofAdults"))

NoofChildreen = cint(Request.Form("NoofChildreen"))

ID = request.QueryString("ID")

rtid = request.QueryString("rtid")

sql = "SELECT * FROM Rooms WHERE RoomID NOT IN(SELECT Rooms.RoomID
FROM          Reservations,RoomTypes,Rooms          WHERE
Rooms.RoomTypeID=RoomTypes.RoomTypeID          AND
Rooms.RoomID=Reservations.RoomID AND #" & FromDate & "# BETWEEN
Reservations.StartDate AND Reservations.FinishDate AND #" & ToDate & "#
BETWEEN  Reservations.StartDate  AND  Reservations.FinishDate  AND
Reservations.ReservationID<>" & id & ")AND Rooms.RoomTypeID=" & rtid & ""

'response.Write(sql)

set conn = dbconnexion()

set rs = conn.execute(sql)

if not rs.eof then

sure = datediff("d",FromDate,ToDate)

sqlroom = "SELECT Price FROM Roomtypes WHERE RoomTypeID=" & rtID

set rsroom = conn.execute(sqlroom)

total =
(sure*rsroom("Price")*noofadults)+(sure*rsroom("Price")*(noofchildreen*0.75))

'response.Write(total)

sqlupdate = "UPDATE Reservations SET NoofAdults=" & NoofAdults &
",NoofChildreen=" & NoofChildreen & ",Total=" & total & ",StartDate=#" &

```

```
sqldate(FromDate) & "#,FinishDate=#" & sqldate(ToDate) & "# WHERE
ReservationID=" & id
```

```
'response.Write(sqlupdate)
```

```
%>
```

```
The reservation is updated. <a href="reservations.asp">Back</a>
```

```
<%
```

```
conn.execute(sqlupdate)
```

```
end if
```

```
%>
```

ROOMS.ASP

```
<!--#include file="includes/admintop.asp"-->
```

```
<%
```

```
set conn = dbconnexion()
```

```
sql      =      "SELECT      *      FROM      Rooms,RoomTypes      WHERE
Rooms.RoomTypeID=RoomTypes.RoomTypeID ORDER BY RoomID"
```

```
'response.Write(sql)
```

```
'response.End()
```

```
set rs = conn.execute(sql)
```

```
%>
```

```
<table      border="1"      cellpadding="0"      cellspacing="0"      width="100%"
bgcolor="#FF9900">
```

<tr>

<td>Room Type

</td>

<td>Picture

</td>

<td>Info

</td>

<td>Longinfo

</td>

<td>Price

</td>

<td>RoomID

</td>

<td>Stair

</td>

<td>Update

</td>

<td>Delete

</td>

</tr>

```

<%

while not rs.eof

%>

<tr>

<td><%=rs("RoomType")%>

</td>

<td><%=rs("Picture")%>

</td>

<td><%=rs("Info")%>

</td>

<td><%=rs("Longinfo")%>

</td>

<td><%=rs("Price")%>

</td>

<td><%=rs("RoomID")%>

</td>

<td><%=rs("Stair")%>

</td>

<td><%if Session("UserLevel")>2 then%>

<a href="roomu.asp?id=<%=rs("RoomID")%>">Update

```

```

</a><%end if%>

</td>

<td>

<%if Session("UserLevel")>2 then%>

<a href="roomd.asp?id=<%=rs("RoomID")%>">Delete

</a><%end if%>

</td>

</tr>

<%

rs.movenext

wend

%>

</table>

<%

conn.close

set conn = nothing

%>

```

ROOMD.ASP

```

<!--#include file="includes/admintop.asp"-->

<%

```



```
ID = request.QueryString("ID")

set conn = dbconnexion()

sql = "DELETE FROM Rooms WHERE RoomID=" & ID

conn.execute(sql)

%>

Room is deleted.<a href="rooms.asp">Back</a>

<%

conn.close

set conn = nothing

%>
```

ROOMU.ASP

```
<!--#include file="includes/admintop.asp"-->

<%

if Session("UserLevel")<2 then response.End()

ID = request.QueryString("ID")

set conn = dbconnexion()

sql = "SELECT * FROM Rooms WHERE RoomID=" & ID

set rs = conn.execute(sql)

%>

<form method="post" action="roomup.asp?ID=<%=ID%>">
```

```
<table bgcolor="#FF9900" align="center">
```

```
<tr>
```

```
<td align="left">
```

```
Stair:
```

```
</td>
```

```
<td>
```

```
<input type="text" name="stair" value="<%=rs("STAIR")%>">
```

```
</td>
```

```
</tr>
```

```
<%=
```

```
sql = "SELECT * FROM RoomTypes"
```

```
set rslevel = conn.execute(sql)
```

```
%>
```

```
<tr>
```

```
<td>
```

```
User Level:
```

```
</td>
```

```
<td>
```

```
<select name="RoomType">
```

```
<%=while not rslevel.eof%>
```

```

<option
value="<%=rslevel("RoomTypeID")%>"><%=rslevel("RoomType")%></option>

<%

rslevel.movenext

wend

%>

</select>

</td>

</tr>

<tr>

<td colspan="2">

<input type="submit" value="Update">

</td>

</tr>

</table>

</form>

<%

conn.close

set conn = nothing

%>

```

ROOMUP.ASP

```
<!--#include file="includes/admintop.asp"-->
```

```
<%
```

```
set conn = dbconnexion()
```

```
ID = request.QueryString("ID")
```

```
stair = request.Form("stair")
```

```
roomtypeID = request.Form("RoomType")
```

```
sql = "UPDATE Rooms SET stair='" & stair & "',RoomTypeID=" & roomtypeID & "
WHERE RoomID=" & ID
```

```
'response.Write(sql)
```

```
conn.execute(sql)
```

```
%>
```

```
Room is updated
```

```
<a href="rooms.asp">Back
```

```
</a>
```

```
<%
```

```
conn.close
```

```
set conn = nothing
```

```
%>
```