

**I Tutor:
Math for 4th, 5th, and 6th Grade**

by

Kinh Bui

Submitted to
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the Requirements
for
the Degree of Bachelor of Science
in Information Engineering Technology

© Copyright 2006 Kinh Bui

The contents of this document are under copyright of the author. It may not be reproduced and distributed in whole or in part without the written permission of the author.

Kinh Bui

Kinh Bui

3/6/06

Date

Annu Prabhakar

Annu Prabhakar, Faculty Advisor

3/7/06

Date

Patrick C. Kumpf

Patrick C. Kumpf, Ed.D. Interim Department Head

3/8/06

Date

**I Tutor:
Math for 4th, 5th, and 6th Grade**

By

Kinh Bui

Submitted to
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Engineering Technology

University of Cincinnati
College of Applied Science

March 2006

Table of Contents

1. Project Description	1
1.1 Problem Statement	1
1.2 Problem Solution	2
1.3 Design Protocols	3
2. User Profile	5
2.1 Administrator	5
2.2 Student	5
2.3 Teacher	6
3. User Requirements	6
4. Current Development	7
4.1 Login Interface	7
4.2 Microsoft SQL Database	7
4.3 Database Table	8
4.4 Database Connection	9
4.5 Security Measurement	9
5. Testing Strategy	10
6. Time Management	11
7. Software Requirements	13
8. Hardware Requirements	13
9. Budget Outline	13
10. Deliverables	14
11. ASP.NET Code Sample	15
11.1 Web Form Control	15
11.2 WebChart Control	16
11.3 Session Object	16
11.4 Embedding HTML Code	19
11.5 Displaying Embedded HTML Code	19
11.6 Beginning a Session	20
11.7 Ending a Session	20
12 Computer Adaptive Testing (CAT)	20
12.1 Applying CAT Logic	21
12.2 Object Details	22
12.3 Auto-Number Generator	23

12.4 Questions from the Database	23
13. Saving a Session	24
14. Loading a Session from the Save Table	24
15. Calculating the Average Skill	26
16. Administrator On-line Tasks	26
16.1 Maintain User List On-line	26
16.2 Assigning Teacher On-line	27
16.3 Question Status On-line	27
16.4 Adding Questions On-line	28
17. User's Games for Leisure	28
17.1 Click-a-Button	29
17.2 Hangman Math	30
17.3 Write-a-Check	31
18. Conclusion	31
19. References	32

Figures and Tables

Figure 1: Login	vi
Figure 2: Sitemap	3
Figure 3: Relationship Diagram	7
Figure 4: CAT Math Session	19
Figure 5: Saved Record	22
Figure 6: Load Saved Sessions	23
Figure 7: Click-a-Button Game	27
Figure 8: Hangman Math Game	28
Figure 9: Write-a-Check Game	29
Table 1: Security Measure	8
Table 2: Fall Schedule Weekly Details	11
Table 3: Winter Schedule Weekly Details	11
Table 4: Details Cost	12

Abstract

"I Tutor" Web site is designed for 4th, 5th, and 6th graders who want to practice or test their skills in math. This Web site is created by using ASP.NET technology for on-line flexibility. Because it is a Web application, eventually anyone can access this application all over the world.

This Web application contains a Computer Adapted Testing (CAT) logic, which is used to draw problems from the database, a server-end data source, for dynamic math sessions. Each session has a set of ten questions and the questions are different each time. The CAT method measures the student's performance based on the correctly-answered questions from each completed session. As the result, the sessions can either be easy, normal or hard. Before starting a math session, users have the option to pick a Teacher, a person who is in charge of a group of students, who they want to review their results. In addition, there is a game and activities room, which includes hangman math, click-a-button, and write-a-check, for a fun and friendlier Web environment. Management of the data is at effortless, because the administrator can perform basic maintenance tasks such as update, delete, and insert information to the database across the Internet.

"I Tutor" Web site is fun and beneficial to those who want to enjoy doing math while surfing the Internet. Education does not have to be limited to school; it can also be implemented on-line for convenient and quick access anywhere in the world. See **Figure 1** for a sample of the login page.



Figure 1: Login

I Tutor: Math for 4th, 5th, and 6th Grade

1. Project Description

“I Tutor” Web site is designed for 4th, 5th, and 6th graders who want to practice or test their skill in math. Because it is a Web application, eventually anyone can access this system to practice and improve math skills. To add some challenges to the application, a Computer Adapted Testing (CAT) method is used to draw problem from the database for a more dynamic Web application. Each session will have ten questions and the questions will be different for each session. Along with each session, users have the option of picking which teacher who they want to review their results.

1.1 Problem Statement

Math is a major subject taught in education systems around the world. We see math from first grade to high school, through college, and finally through our daily work life. Strong, basic math skills help increase the productivity of daily life. For example, every student has to take some kind of math in college to fulfill the curriculum. Many companies required applicants to have basic math skills in order to join their workforce. Because math is globally applied, competition among nationalities is unavoidable.

Many students in the United States have lower math skills compared to European and Asian countries. Thomas C. O'Brien, author of “Parrot Math”¹, provides the following facts about mathematics in countries around the world.

¹ <http://www.pdkintl.org/kappan/kobr9902.htm>

- In the first International Study of Achievement in Mathematics, published in 1967, American 13-year-olds finished next to last among 10 major industrial nations.
- In the Second International Mathematics Study, conducted during the 1981-82 school year, American eighth-graders ranked 10th of 20 national groups in arithmetic, 12th in algebra, 16th in geometry, and 18th in measurement. In all subtests, America's overall scores were at or below the median for the entire group.
- In a comparison of 14 national groups of 9-year-olds reported in 1992, American children came in next to last, besting only Slovenia. In the same research, American 13-year-olds tied with Spain for next-to-last place and bested only Jordan.
- TIMSS is the largest, most comprehensive, and most rigorous international comparison of education ever undertaken. During the 1995 school year, the study tested the math and science knowledge of half a million students from 41 nations. In mathematics, U.S. eighth-graders posted scores in the middle of the pack, slightly below the international average (O'Brien).

Nationally, we hear many proposals by the President or governors to standardize the math skill level; passing the proficiency test is one way to measure the performance for higher education. Math is one of the required subjects that all schools need to pass in the "No Child Left Behind Act" proposed by President Bush when he ran for office. Now this act applies nationally.

1.2 Solution

We as a nation want to improve our children's math skills, beginning with the basic. "I Tutor" Web application for 4th, 5th, and 6th graders is a supplemental to any elementary students who want to practice and improve his/her math skills. This Web application, using Active Server Page .NET (ASP.NET) technology, which works well with many standard Web browsers, will be dynamically accessible by students who are interested in practicing math or testing their math skills. It can also track the student's performance and then

group the student into Easy, Normal, or Hard skills math groups. All the questions are grouped by the administrator. However, the student and teacher can submit comments for any changes. Many believe that the best way to learn something is by actually doing it. By combining Hypertext Markup Language (HTML) and Active ASP.NET technologies with a relational database that is hosted on the senior server, this math Web application is available to improve education for all intended users.

1.3 Design Protocols

The entire Web interface has a fade green to white radian color template. The navigation menu is located on the left side by the division line. On the top of the page is the general title area. On the center top area, there is a logout and a welcome message. An error warning label is in red with appropriate message; a red star located next to the input field indicates the same message. Otherwise, a blue label text message is used to indicate a successfully implementation to a requested action. All hyperlinks and action buttons have a blue text to indicate a clickable action. **Figure 2** shows an abstract sitemap for this Web Application.

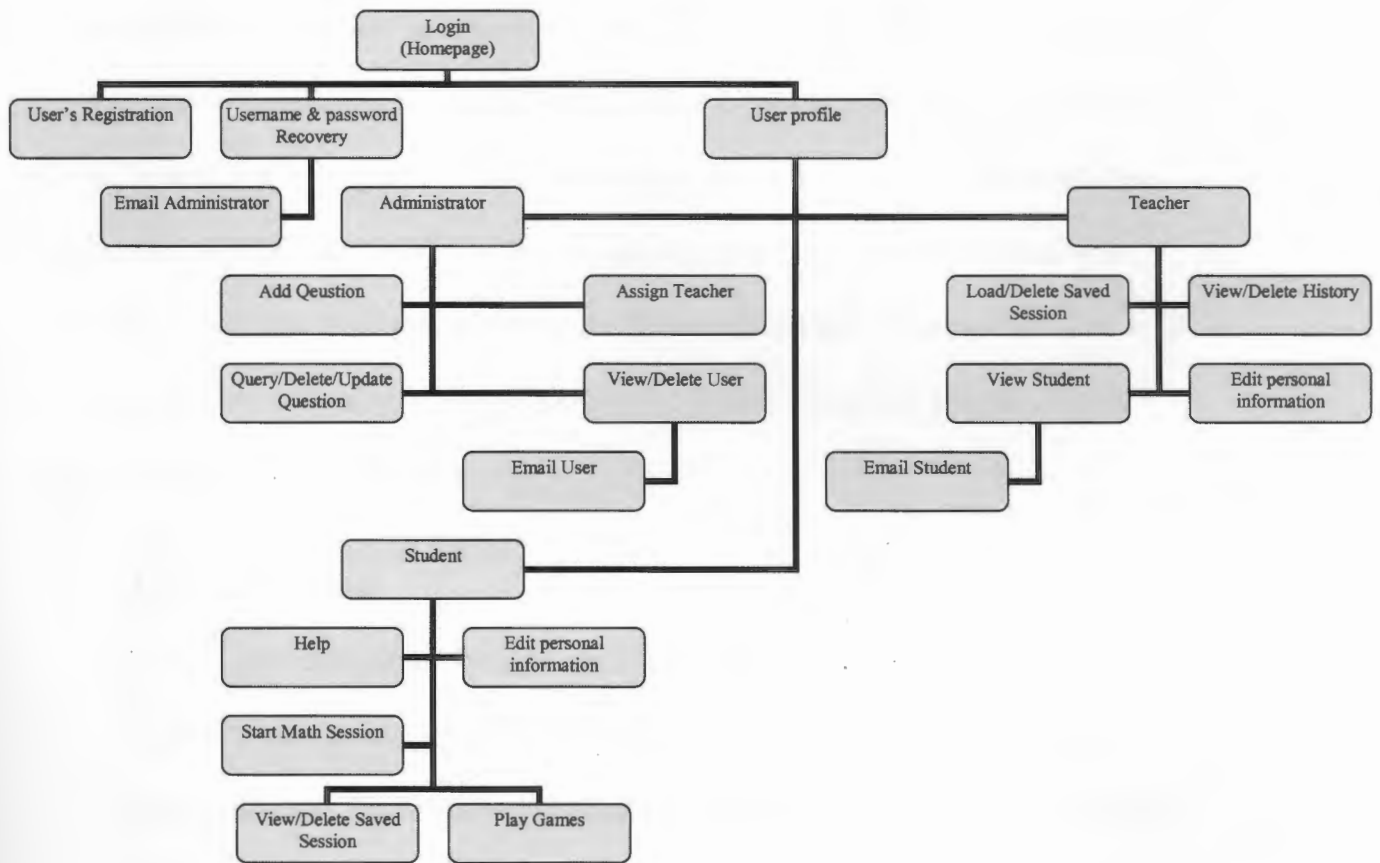


Figure 2: Sitemap

2. User Profile

This Web application has three different users. The administrator is a person who will be maintaining the server-end tasks; the 4th, 5th, or 6th grader will be accessing the application through the Internet to perform math; finally, the teacher will be managing the student groups. The administrator is expected to be familiar with database maintenance issues and will be expected to keep the Web-application active at all times. Each student is expected to be familiar with Internet Explorer or other browser for the Webs. In addition, each student needs knowledge of basic math skills such as addition, subtraction, multiplication, and division in order to perform the problem in this Web application. All presented problem are in the format of multiple choices, true/false, or fill in the blank format. Each user's type has specific roles to carry out. Details of each user's role are listed below.

2.1 Administrator

The administrator will be maintaining the server-end tasks, which include Web site maintaining, updating the database, and providing feed back to any user's requests. The following on-line update tasks to the database are removing users, removing questions, and assigning teachers.

2.2 Student

The 4th, 5th, or 6th grader is the student who will be accessing the application through the Internet to practice math problems or test their math skills. Every completed math session will be saved for skill calculation. These records can be deleted anytime by clicking on the "View History" link from the user homepage. When opening a math session, the student can save that session by

clicking on the “Save Session” button. All saved sessions are loadable by clicking on the “Load Saved Session” link from the user homepage. Like the history record mentioned above, all saved sessions are removable from the “load save” page. Each user has the option to change his/her personal information anytime he/she desires. By clicking on the “Update Profile” button, the user can change first and last name, email address, or password. The student can also enjoy some of the popular games implemented in a math format while visiting the Web site. Please see session 17 for more details.

2.3 Teacher

The teacher has all the functionality the student has plus more. The teacher has access rights to view the list of all his/her students that have histories of successfully completed any math sessions the student did previously. This list provides status of how the students perform. The “User Group” column on this list allows the teacher to select and view the selected student’s result distribution on a column chart. Based on the report, the teacher can email the listed students for better communication. The teacher also has the ability to create quick worksheets for practice purposes.

3. User Requirements

The administrator is expected to be familiar with database maintenance and expected to keep the Web application accessible at all times. The teacher must be familiar with a Web browser. The teacher is also expected to have at least one student so he/she can review the student results. Each student is expected to be familiar with a standard Web browser; however, the Microsoft Internet Explorer 6.0 (IE) is

recommended. The student needs basic math skills such as addition, subtraction, multiplication, and division in order to perform most of the problems; basic reading skills can really help analyzing word problems. All presented problems will be in the format of multiple choices, true/false, or fill in the blanks.

4. Current Development

For the winter quarter of 2005, the focus of this Web application is on the design of a user-friendly Web interface and a functional CAT algorithm that accesses a relation database for its data source.

4.1 Login Interface

Logging in this Web application is required. Each user profile will be updated and stored in the database regularly, which will be referenced throughout the application. For example, the teacher needs to know all students who are under his/her class. Also, when each student saves a session, he/she must use the student ID.

4.2 Microsoft SQL Database

Because all of the data will be drawn from a database, there must be a reliable database to give the demanding Web application access to the questions and the user's data. Microsoft Structured Query Language (SQL) Database 2000 is one of the trusted databases for the Web application. The database contains information about the user's login credentials, personal profiles, and math problems. The administrator will be the only one who can carry out the changes to the database. See **Figure 3** below is the database schema.

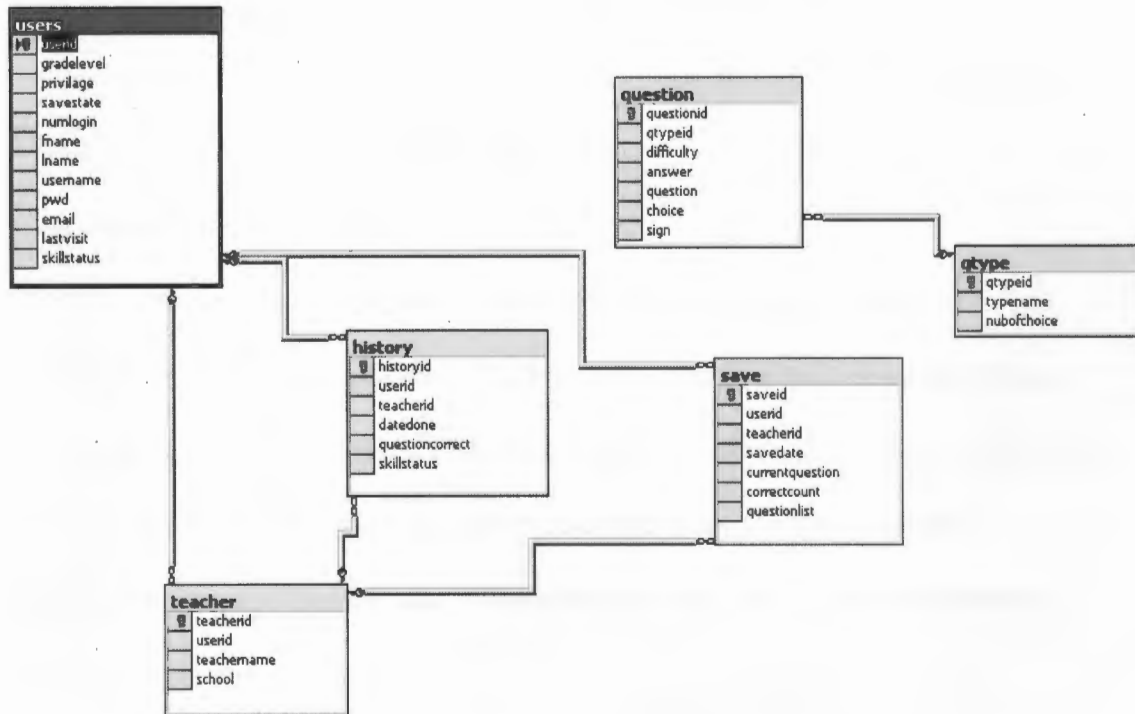


Figure 3: Relationship diagram

4.3 Database Tables

All users are located in the users table. The related tables to the users' table are save table, history table, and teacher table. A save record is added when a user save an unfinished math session. A history record is added to the history table automatically when a user completed a math session. It is auto generated without the user's notice for skill calculation purposes. A teacher record is added only when the administrator updates a user into teacher role. The question table and qtype table are related only because qtype contain question type such as multiple choices, true/false, and fill in the blank question. The only relationship between question and a user is that in question table, there is a difficulty column; it contains a range of number from one to nine. The first three numbers belong to

all 4th grade users, the next three numbers belong to 5th grade user, and the last three numbers belong to 6th grade user. This is how the application logic separate which question belongs to which user group.

4.4 Database Connection

Connection to a database table for information is frequently used throughout this Web application. Therefore in order to avoid repeated coding, a custom class is made. MakeConnection class is a custom class made to serve any connection call to the database. Other custom classes are SessionState, WriteNumber, and GameStatus. Please see the attached CD for more details about these custom classes.

4.5 Security Measurement

To guard against data corruption from user inputs, secure measurement is a required feature that will guarantee the life of the application and its data. This Web application will take care of data inputs from the moment of registration until each logoff or resign from the user's list. The following table shows areas that require a security measure.

Areas	Security	Description
Data Type	Maximize Storage	This gives efficiency to the application when the project grows
Text Fields	Specific String Length	This guards against possible overload input
Password	Encryption	This guards against unwanted users who can potentially login and modify a user's information
Table Relationship	Data Redundancy	This guards against repeating data and ensures good data for each and every query.

Table 1: Security Measure

By the end of winter quarter, this Web application will be completed with error-free pages. This measurement can bring the application to the end of coding. Following the measurement is the following up with user's responses on the performance issue as more and more users accessing this Web application later on.

5. Testing Strategy

Testing is crucial process for any project whether it is big or small. The best strategy in testing a code-project is to perform periodically throughout the development process. The following list details the testing phases performed so far.

1. Compare each input field with the assigned table's field
2. Review proper coding for each query made
3. Review CAT algorithm logic
4. Review user interface such as navigation, images, and colors
5. Validate the input fields such as email address and names
6. Validate login and logoff procedure
7. Validate valid user
8. Test the application with actual users including 5th and 6th graders, myself, and Professor Prabhakar, the technical advisor

6. Time Management

The project requires time as well as effort because it has considerable research criteria, challenging programming methods, multiple testing phases, and documentations. The first quarter of senior design was spent researching and getting to know the equipment mentioned in **Table 2**. The second quarter of senior design was focused on implementing the database, gathering test data, developing .aspx pages, and documenting the findings. Finally, the third quarter of senior design includes more testing, making changes from the test results, making changes to the database, and finalizing the written parts of the documentation. The following timeline contains a schedule of specific tasks in a timely manner.

Fall 2005 Timeline		
Task	Time	Status
Research GUI	09/23-09/30	Completed
Implement GUI	10/1-10/16	Completed(<i>changes apply through the quarter</i>)
Implement Database	10/1/-10/16	Completed(<i>changes apply through the quarter</i>)
Implement Secure Login Web site	10/18-11/5	Completed(<i>changes apply through the quarter</i>)
Programming CAT Logic	11/6-11/15	Completed(<i>changes apply through the quarter</i>)
Testing the Interface	11/16-12/21	Completed(<i>changes apply through the quarter</i>)
Drafting a Design Freeze	10/25-11/05	Completed

Table 2: Fall Schedule Weekly Details

Winter 2006 Timeline		
Task	Time	Status
Finalizing a Design Freeze	Week 1-2	Completed(<i>changes apply through the quarter</i>)
Finalizing CAT Logic	Week 2-4	Completed(<i>changes apply through the quarter</i>)
Loading Data to the Database	Week 2-4	Completed(<i>changes apply through the quarter</i>)
Testing CAT Logic	Week 5	Completed(<i>changes apply through the quarter</i>)
Finalizing all Documents	Week 6	Completed(<i>changes apply through the quarter</i>)
Testing the Application	Week 7-13	Completed(<i>changes apply through the quarter</i>)

Table 3: Winter Schedule Weekly Details

7. Software Requirements

There are numerous types of software needed to carry out the project. The following list contains all software needed for the Web application.

1. Visual Studio .NET 2003
2. Windows Server 2000 with ASP.NET 1.1 technology
3. Microsoft SQL Server 2000
4. Microsoft Internet Explorer 6.0(recommended)

8. Hardware Requirement

Hardware is also required to carry out this Web application successfully. The following list contains all the required hardware.

1. Microsoft IIS Web Server
2. Microsoft SQL 2000 Server
3. Client computer with Internet access

9. Budget Outline

The following software and hardware list gives an estimated cost for the project. Because it is a Web application, I need a Microsoft IIS Web server, a Microsoft SQL Server 2000 database, programming technology, and photo editing tools. **Table 4** outlines all of the equipment costs for this project.

Tools	Cost
SQL 2000 Server Standard	\$300.00
Macromedia Flash MX 2004	294.98
Adobe Photoshop CS	552.50
Visual Studio .NET 2003	599.95
Windows Server 2003 Standard(5 Cal)	999.00
PC	1,500.00
Total Cost	\$4246.43

Table 4: Details Cost

10. Deliverables

The Web application contains many features such as a user interface, dynamic method to draw questions, a security measurement to guard against data corruption, and games for the leisure. Below is a list of deliverables that are part of this Web application.

1. A single point of authentication
2. A user interface is in ASP.NET for dynamic pages
3. A general navigation for each type of user.
4. Registration for new users
5. Registration for teachers
6. Tasks for specific users.
 - a. Administrator is in charge of managing the database, loading and maintaining questions, assigning users to be teachers, removing users, and responding to user's requests.
 - b. Teacher is in charge of monitoring the student's performance based on the student history, maintaining his/her history and saved sessions, beginning a math session, emailing the student and creating worksheets.

- c. Student is able to begin a new math session, save unfinished math sessions, save a record for any successful session, and play games.

11. ASP.NET Code Sample

With the assistance of Visual Studio.NET 2003, ASP.NET coding project becomes much easier in term of time and effort putting into each *.aspx* page. All coding are done in C#; the coding files are separated from the HTML files for readability purposes. All of the coding pages contain Web form controls, controls that are processed by the server before sending back to the client's browser. Including all the Web form controls, session objects, response objects, and embedded HTML codes, the developer gains more control over ASP.NET pages. For the users other then the administrator, they can enjoy the flexibility ASP.NET provides.

11.1 Web Form Control

The Web form controls include hyperlink button, button, dropdown box, and radio buttons that are containing methods and events in which the server processes all the code before sending back the data to the client's browser. The snippet below shows how to code a Web form control from the HTML file.

```
<asp:Button ID=btnSignIn Runat="server" Text="Sign In"
onClick="btnSignIn_Click" />(Kalata)
```

All asp.net server controls start with a `<asp:...` and end with `/>`. The attribute `Runat='server'` indicates to the server that it must process from the server first.

11.2 WebChart Control

WebChart control is a server control, a free download from the Carlosag Web site, which allows the user to create a graph at run time. WebChart control uses to create a distribution column chart of the user's results from all the completed math session. This column chart allows the teacher to understand why certain users categorized into the user group. To use this control, the developer must add a reference to the project solution. Once that is done, the developer must also add the control to the toolbar. To add this control to the toolbar, please right-click on the toolbar and select "Add/Remove Items" item menu. Once a small dialog shows up, click on the "Browse" button to browse to the .dll file to add. For the final inclusion, add a "using WebChars" namespace to the project using at the top of the project file. Please the "details.aspx.cs" page for more coding help.

11.3 Session Object

Another server object that is used throughout the .aspx pages is the session object. A session object is a session uniquely identified for each user who uses the Web application; each has a unique session ID. Therefore, each session contains a collection of session variable. The snippet below shows how to use the session object from a C# source file.

```
//assign a new variable in the session object  
this.Page.Session[<variablename>] = <variable value>;  
  
//reading an existed string variable in the session object  
String str = this.Page.Session[<variablename>].ToString();  
  
//remove an existed session variable  
This.Page.Session.Remove("<variablename>");(Kalata)
```

All session variables exist as long as the browser opens and are active in 20 minutes. An example of session variable used in the Web application is *MakeConnection* object. *MakeConnection* is a class that contains properties and methods that communicate to a database. Its properties and methods include Dataset, Username, Password, Role, Grade, Pool1, Pool2, Pool3, SqlConnection, SqlQuery, GetData, CheckValidUser, InsertQuestion, and GetQuestion. Once an instance of the *MakeConnection* class is created, it can be stored as an object in the session object. The information can then be retrieved and used throughout the current session. Please consult the attached CD for more information about *MakeConnection* class. The snippet below is a small sample of the *MakeConnection* class.

```

using System;
using System.Data;
using System.Data.SqlClient;
using System.Collections;
using System.Threading;

namespace SeniorDesign
{
    public class MakeConnection
    {
        //global variables////////////////////////////////////
        private string strconn;
        private string strquery;
        private string u;
        private string p;
        private string r;
        ArrayList p1;
        ArrayList p2;
        ArrayList p3;
        private string grade;
        private SqlDataAdapter da;
        private DataSet ds;
        //end global variables////////////////////////////////////

        public MakeConnection()
        {
            strconn = "server=DIT-SENIOR;Initial
Catalog=buikl;uID=bui;pwd=bui;";
            da = null;
            ds = new DataSet();
            u = "";
            p = "";
            p1 = new ArrayList(); //Easy pool
            p2 = new ArrayList(); //Normal pool
            p3 = new ArrayList(); //Hard pool
            grade = "";
            r = "";
        }
        public DataSet Ds
        {
            get
            {
                return ds;
            }
            set
            {
                ds = value;
            }
        }
    }
}

```

11.4 Embedding HTML Code

Using Web form controls and session object variables may not be enough when dealing with an ASP.NET Web page. Embedding HTML code to the project increases the option of creating a dynamic ASP.NET page. In the Web application, *StringBuilder* class is used to encapsulate HTML string values before sending back to the requested user. The snippet below shows how to encapsulate HTML code inside an instance of the *StringBuilder* object from a C# source file.

```
StringBuilder sb = new StringBuilder();  
sb.Append("<input type='textbox' name='anyname' value='anyvalue'  
>");(Darie and Watson)
```

11.5 Displaying Embedded HTML Code

In order to display the above information to the user's browser, another server object is being applied. It is the *this.Response* object, which can return HTML information to the browser. The snippet below shows how to send HTML code to the user's browser from a C# source file.

```
this.Response.Write(sb);
```

Without assigning the textbox above any specific location, the textbox will show up in the top-left corner of the current page. Any absolute positioning in HTML code can be used a style sheet to define its position in a class or inline style. Style sheet is the file containing specific HTML page format such as background color, forecolor, font family, or absolute position. For example, all tables in the application are using inline style to specify its absolute position on the HTML page. Please consult the attached CD for more samples of the *this.Response* object that used HTML code and inline style.

11.6 Beginning a Session

A session begins when the user logs into the Web application. Session objects keep information about the current user throughout the Web application pages. For example, a *SessionState*, a *MakeConnection*, and others HTML embedded session objects are kept available for ready to use. The advantage of keeping these objects alive global wise is because to reduce the coding redundancy and enjoy the quick access.

11.7 Ending a Session

To create an up-to-date record about the user who logging in Math Tutor Web site, a date stamp is logged for the user. A counter keeps tally of how many time the user visited the Web site. This record-keeping status is nothing more than a status to the user. No information is used against the user. It is advised that the user is to login and logout before closing the browser.

12. Computer Adaptive Testing (CAT)

The CAT is an algorithm that provides a dynamic way to gather data from the database and present it back to the user; this algorithm categorizes into groups based on the performance. The question from the SQL database is the data source for the CAT algorithm. The CAT algorithm begins with ten random questions from each of the three question categories. The question categories are Easy, Normal, and Hard. Based on the previous correct answer, the CAT algorithm provides a harder question to the student. Otherwise, CAT algorithm returns an easier question. The session will go on until all ten questions are answered. Instead of a percentage on how well the student performed, CAT groups the result into levels of math skills that the student acquired based on the question

answered. All of the difficulty levels of the questions are evaluated based on the general survey with the elementary teachers and the corresponding math books. The following math books are used: Saxon Math 54, How Well Does Your Child Do Math?, How to Develop Your Child's Gift and Talent in Math, and Hands-On Math. The more sessions the students completed the more accurate the result will be in grouping. **Figure 3** shows a user interface of the Web's CAT algorithm that is drawn a generic question at runtime.

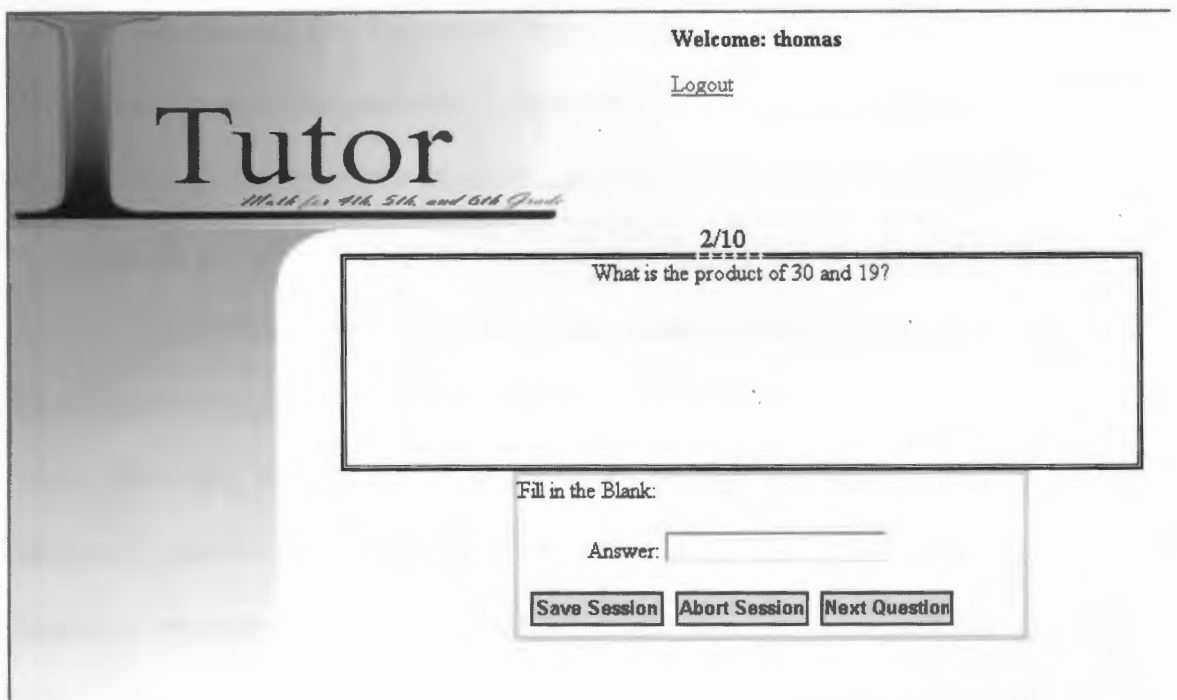


Figure 4: CAT Math Session

12.1 Applying CAT Logic

The idea of CAT logic is as follows. If the current problem is correctly answered, then the next question is of a higher level than the current question. However, if the question is answered incorrectly, then the next question is of a

lower level than the current question. In a way, CAT makes decisions in which type of question the user should do. By using this logic, the problem presents the following objects in order to compute and carry out the problem: three Array List objects, one *SessionState* object, three correct counters, and one question counter.

12.2 Object Details

The three Array Lists are objects that contain the collection of question ID, which is uniquely identified in the database table question. The first list contains easy question ID. The second list contains a normal question ID. The last list contains the hard question ID. Each list can have a maximum of ten question IDs. All question IDs should be randomly selected before adding them to the appropriate list.

A *SessionState* object is a custom class created for the purpose of returning everything there is to know about the current question that the student is using. This object should query a question from the database based on the question ID provided from one of the three Array Lists. This object returns a dataset that contains the question details, which includes the actual problem, answer, choices, and type of question.

The three correct counters are counters that keep a tally of each level of question when the user answers correctly. So at the end of each session, an easy level should have a counter value. The same is applied to normal and hard level counter.

The last counter in a session is the question counter. How many questions are already used during the session? This counter should be less than or equal to

ten since each session has ten questions. Please consult the *userprofile.aspx.cs* file from the attached CD for coding details about the new math session.

12.3 Auto-Number Generator

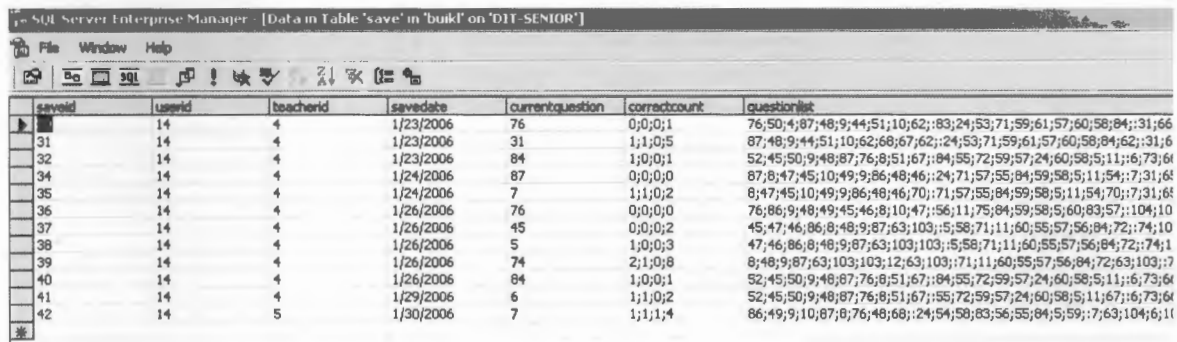
In addition to the CAT algorithm, there is another special feature that is included into the logic. It is the auto-number generator. For any generic question created and stored in the database, the auto-number generator will provide a set of number and answer to that question. For example, “what is the sum of #, #, #?” is a generic question. In this case, the auto-number generator will find all pound sign characters and replace them with integer values. The answer is based on those numbers. All generic questions are marked with one of the following characters: +, -, x, /, #, and w.

12.4 Questions from the Database

Each time the user starts a math session, the questions are randomly drawn from the *question* table from the *buikl* database. In order to complete the CAT logic with error-free, the *question* table must have 30 or more reserved questions for each of the grade level. The question must be divided into three categories; ten for easy, ten for normal, and ten for hard level. If the session takes longer than usual to start, then it is the sign of insufficient question from the *question* table. The user is advised to contact the administrator to resolve the problem as soon as possible.

13. Saving a Session

Saving a session during any given point in a math session is one of the deliverables for this project. In order to save a session, the information described in session 12 is required for this to function correctly. The page redirects itself to the user homepage after clicking on the “Save Session” button provided on the current page. Each saved record from a session requires to have all question IDs from different lists, current question ID from the *SessionState* object, the correct counter from three levels, the question counter, the current user’s ID, and the user’s teacher’s ID. All these pieces must be in string formation in order to save a record to the *save* table. Please see the *mathsession.aspx.cs* file from the attached CD for coding details about saving a session. Figure 5 provide a sample of what each save session looks like from the database.




saveid	userid	teacherid	savedata	currentquestion	correctcount	questionlist
31	14	4	1/23/2006	76	0;0;0;1	76;50;4;87;48;9;44;51;10;62;83;24;53;71;59;61;57;60;58;84;31;66
32	14	4	1/23/2006	31	1;1;0;5	87;48;9;44;51;10;62;68;67;62;24;53;71;59;61;57;60;58;84;62;31;6
34	14	4	1/23/2006	84	1;0;0;1	52;45;50;9;48;87;76;8;51;67;84;55;72;59;57;24;60;58;5;11;6;73;6
35	14	4	1/24/2006	87	0;0;0;0	87;8;47;45;10;49;9;86;48;46;24;71;57;55;84;59;58;5;11;54;7;31;6
36	14	4	1/24/2006	7	1;1;0;2	8;47;45;10;49;9;86;48;46;70;71;57;55;84;59;58;5;11;54;70;7;31;6
37	14	4	1/26/2006	76	0;0;0;0	76;86;9;48;49;45;46;8;10;47;56;11;75;84;59;58;5;60;83;57;104;10
38	14	4	1/26/2006	45	0;0;0;2	45;47;46;86;8;48;9;87;63;103;5;58;71;11;60;55;57;56;84;72;74;10
39	14	4	1/26/2006	5	1;0;0;3	47;46;86;8;48;9;87;63;103;103;5;58;71;11;60;55;57;56;84;72;74;1
40	14	4	1/26/2006	74	2;1;0;8	8;48;9;87;63;103;103;12;63;103;71;11;60;55;57;56;84;72;63;103;7
41	14	4	1/26/2006	84	1;0;0;1	52;45;50;9;48;87;76;8;51;67;84;55;72;59;57;24;60;58;5;11;6;73;6
42	14	5	1/29/2006	6	1;1;0;2	52;45;50;9;48;87;76;8;51;67;55;72;59;57;24;60;58;5;11;67;16;73;6
42	14	5	1/30/2006	7	1;1;1;4	86;49;9;10;87;8;76;48;68;24;54;58;83;56;55;84;5;59;7;63;104;6;11

Figure 5: Saved Record

14. Loading a Session from the Save Table

At some point, the user will access the saved sessions and eventually wants to reload and continue where he/she left off. Therefore, the reverse of “Save a Session”,

mentioned in session 13 is true. Each saved record of a session required to have all the question IDs from different lists, the current question ID, the correct counter from three levels, the question counter, the user ID, and the user's teacher's ID. Therefore, these objects must be recreated and reassign to the corresponding objects to the appropriate column from the *save* table. All of these objects must then assign back into the session objects. Please see the *loadsavesession.aspx.cs* file from the attached CD for coding details about loading a saved session. **Figure 6** shows a sample of saved sessions the users have in his profile.



Tutor
Math for 4th, 5th, and 6th Grade

Record(s) Delete Record

34 ▼ ▼

>> <<

Delete

Welcome: thomas

[Logout](#)

Load	Date Saved	Question Description
34	1/24/2006	What is # - # = ?
35	1/24/2006	There are 5 red marbles, ...
36	1/26/2006	Tom has 10 marbles, Harry...
38	1/26/2006	There are 123 blue birds ...
41	1/29/2006	During the Summer break...
42	1/30/2006	There are 5 red marbles, ...
43	2/2/2006	You run 3 miles and your ...
44	2/3/2006	What is # + # = ?
45	2/6/2006	You have 100 pounds of Ro...
46	2/6/2006	100 = 10 * 11
51	2/15/2006	The quotient of 120/5 is 2...
55	2/21/2006	Comparison: 33 feet < 3...
56	2/21/2006	Tom has 10 marbles, Harry...

Figure 6: Load Saved Sessions

15. Calculating the Average Skill

Calculating the student's performance is an important deliverable for this project. In order to calculate the skill level for each student, one or more completed math session must be saved in the *history* table. For each history record, there is a column that contains all three correct counters of each question level. These numbers are the main ingredient for calculating the average skill level. Each number counter is summed up separately and divided it by the number of completed sessions. Whatever average comes out the highest is the current average skill level for this Web application. Please see the *mathsession.aspx.cs* file from the attached CD for coding details about computing the average skill.

16. Administrator On-line Tasks

The tasks that the administrator can perform on-line are editing users, editing questions, assigning teachers, and emailing all users. These tasks allow flexibility and availability on-line maintenance to the Web application. Any details about the coding methods please see the attached CD for more details according to the pages mention below.

16.1 Maintain User List On-line

The *userlist.aspx* page contains both deletion and email users. The clickable hyperlink on the Delete column of this page's table is the delete feature. Once click the ID hyperlink, the data then submits to the *deleteuser.aspx* page and performs the deletion. Right after the deletion, the page redirects back to the *userlist.aspx* page with the updated list of the remaining users. At the same time,

the administrator can also click on any hyperlinked email address on the address Email column to email the selected user.

16.2 Assigning Teacher On-line

Assigning a user into a teacher's role is a simple task. Any existed user can become a teacher who then manages a group of student just by changing the *privilege* field from the *users* table. The *assignteacher.aspx* page shows non-teacher users in a dropdown box. The administrator can select any available user with a school name and then click the "Add Teacher" button to upgrade the user's role. It is a simply click away to grant someone a more responsible role in this Web application.

16.3 Question Status On-line

Editing questions from the database can also be performed on-line. The page *ViewQuestion.aspx* contains ways to query the desired group of questions. On this page the administrator can perform two tasks: delete and update a question. The Delete column contains all hyperlinked question ID for deletion. The clicked hyperlink transfers to the *QuestionDelete.aspx* page and then performs the deletion. Again, the page redirects itself back to the *ViewQuestion.aspx* page once the operation is done. On the same page, the administrator can also click on any hyperlinked ID from the Update column to update a question. Any selected update hyperlink then transfers to the *QuestionUpdate.aspx* page to let the administrator perform the necessary editing.

16.4 Adding Questions On-line

Finally, the administrator can add questions to the database right away through the Internet. The page *questiontype.aspx* lets the administrator select a type of question he/she wants to add. After submitted the question type, the page redirects to the *addquestion.aspx* page automatically. Each question type has its own question template. The *addquestion.aspx* page shows the appropriate template for the new question.

17. User's Games for Leisure

This feature is an addition to the Web application, which includes Click-a-Button, Hangman Math, and Write-a-Check game activities. Their purposes are for relaxation and leisure while doing the actual math. This feature allows more choice to do after logging into the Web application.

17.1 Click-a-Button

Click-a-Button is a game that the user must evaluate the provided math problem and then click a button that matches the result of the provided math problem. There are 25 problems each and the user can only allow ten mistakes before the game is over. The game is aggressively changed from integer numbers to decimal and then to negative decimals when the student scores 9 or fewer mistakes from each new game. See **Figure 7** for a sample of the Click-a-Button game.

48	52	33	88	74
53	24	68	42	51
79	39	48	20	16
58	2	53	44	76
52	80	21	45	38

Status:

Count Wrong: 0

$$\begin{array}{r} 28 \\ + \\ 25 \\ \hline \end{array}$$

Figure 7: Click-a-Button Game

17.2 Hangman Math

Next game is the Hangman Math. This is similar to the classic hangman game where the user makes guesses. The only difference in this game is the game's objective; it is to find the missing number from the presented problem. All problems are randomly generated every time the game loads. The game only allows five mistakes or when the hangman figure is fully drawn. Again, the game is aggressively changed from integer numbers to decimal and then to negative decimals when the student scores 6 or fewer mistakes from each new game. See **Figure 8** for a sample of the Hangman Math game.

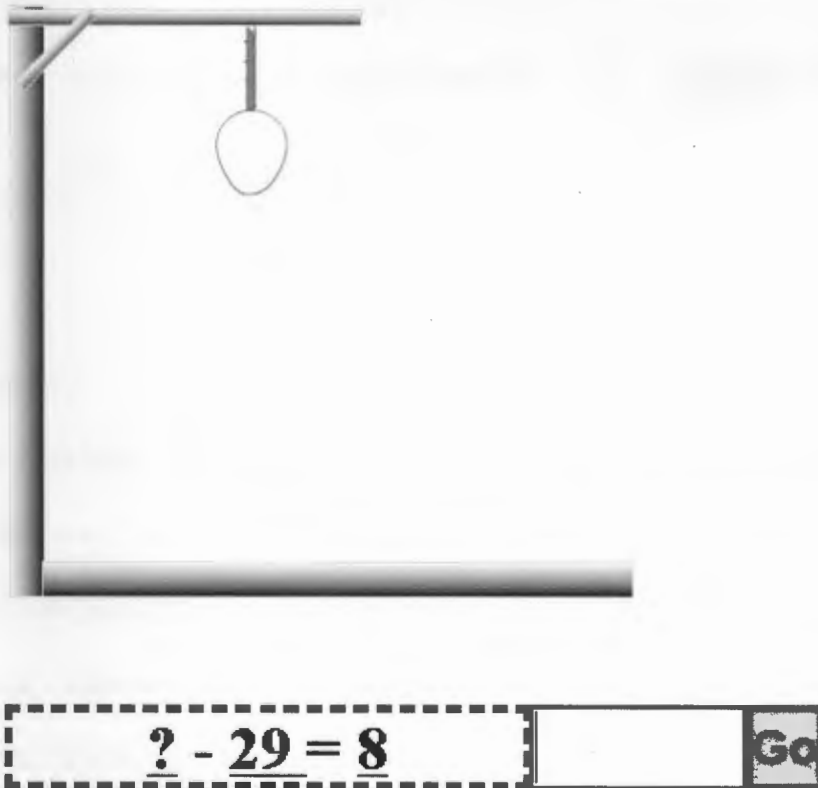


Figure 8: Hangman Math Game

17.3 Write-a-Check

Lastly is the Write-a-Check game. This game's objective is to master writing a numerical number into word or from word to a numerical number. The possible numbers are from 1 to 9999, which is randomly generated every time the game loads. See **Figure 9** for a sample of the Write-a-Check game.

Thomas Thomas 64567
1234 My Street Dr.
Cincinnati, OH 45202

2/24/2006
Date

Pay a check picture \$ 5887
to the order of Dollars

MY BANK Memo:
:00 11111 11 :. 2222 123456789"

Please enter word phrase that matches with the given value

Figure 9: Write-a-Check Game

18. Conclusion

This Web application provides a way for elementary student to practice math problems little or no cost. This Web application provides a way to measure a student's performance based on all the sessions he/she performed. By doing the problems, learning and improving basic math builds a stronger foundation for the future challenges in the world of mathematics. Don't avoid it, do it.

References

1. "ASP.NET". 2005. Mindcracker LLC. <http://www.csharpcorner.com/> 09-12-2005.
2. Carlosag.net. 2006. "WebChart Control for ASP.NET." <http://www.carlosag.net/Tools/WebChart/Default.aspx>. 2-12-2006.
3. Cook, Ann. "Assignments". How Well Does Your Math? U.S.A, Book-mart Press, 1997
4. Daniel, Cristian and Karli Watson. Beginning ASP.NET 1.1 E-Commerce: from Novice to Professional. USA, 2005.
5. Kalata, Kathleen. Introduction to ASP.NET 2nd ed. Cannada, 2005.
6. MathStories.com. 1999-2003. http://www.mathstories.com/Book_17_3pigs_grades23.htm 10-15-2005.
7. Hake, Stephen and John Saxon. Math 54: An Incremental Development 2nd ed. U.S.A, Saxon inc. 1995.
8. Thompson, Frances M. Hand-On Math. U.S.A, West Nyack, NY: 1994.
9. Thomas, O'Brien C. "Parrot Math" <http://www.pdkintl.org/kappan/kobr9902.htm> 11-08-05/
10. Yablun, Ronn. How to Develop Your Child's Gift and Talent in Math. U.S.A, Lowell House, LA: 1995.