

Fall 8-1-2004

ITT Tutoring System

Scott Richards
Dakota State University

Follow this and additional works at: <https://scholar.dsu.edu/theses>

Recommended Citation

Richards, Scott, "ITT Tutoring System" (2004). *Masters Theses*. 39.
<https://scholar.dsu.edu/theses/39>

This Thesis is brought to you for free and open access by Beadle Scholar. It has been accepted for inclusion in Masters Theses by an authorized administrator of Beadle Scholar. For more information, please contact repository@dsu.edu.

RG 35
054
51
Box 1001

ITT Tutoring System

Scott Richards

A project submitted in partial fulfillment of the requirements for the

Master of Science in Information Systems

Dakota State University

2004



MSIS
PROJECT APPROVAL FORM

Student Name: Scott Richards

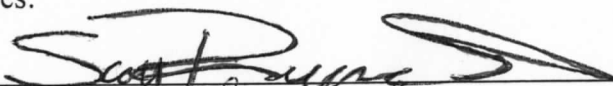
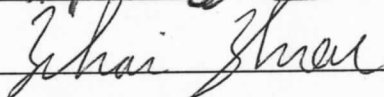
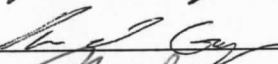
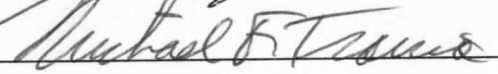
Expected Graduation Date: 8/04

Master's Project Title: The ITT Technical Institute Tutoring System

Date Project Plan Approved: 11/03

Date Project Coordinator Notified and Grade Submitted: _____

Approvals/Signatures:

Student: <u></u>	Date: <u>8/21/04</u>
Faculty supervisor: <u></u>	Date: <u>8/17/04</u>
Committee member: <u></u>	Date: <u>8/17/04</u>
Committee member: <u></u>	Date: <u>8/25/04</u>

*To Ensure Certification of Completion:
Student must bring or send the original to the Graduate Programs Office.
Copies on acid free paper go to the library with the reports for binding.*

Original to Graduate Programs Office
Acid-free copies with written reports to library
Copies to: Project supervisor and committee and to MSIS Coordinator

ABSTRACT

This report details the process of designing and developing a tutor system for the ITT Technical Institute located in Norwood, Ohio. A system request was initiated by the administrative Dean, Mike Traina (See Appendix A). The proposed system is to create a tutoring website. Within this system, tutees will be able to obtain additional information about tutoring; search for available times, courses and tutors; and rate tutoring sessions. Tutors will be able to log tutoring sessions and access training materials.

The system will provide many intangible goals. Faculty will be able to devote more of their time to their classes and less outside-of-class time assisting students. Students will have a more reliable and efficient method of arranging for tutoring. Tutoring schedules will be better utilized and tutors will receive better training. The system extends and improves services to the student population. The long-term goal of the system is increased retention and increased student/faculty satisfaction.

The system will maintain the current schedule showing open times, tutors, courses, and already scheduled appointments. Also within the system will be the session logs showing time and date of each session, length of session, and a transcript summary of the session. This data will be used for tracking and reporting purposes. Finally, the system will contain training documents, tutoring information and other useful tools for both student and tutor. The underlying technology used to support the system will be Apache Web server and MySQL running under the Windows 2000 operating system. The following languages will

be used to develop the interactive components: PHP, HTML, CSS, and SQL. MySQL will be the database of choice for the back-end.

AKNOWLEDGEMENTS

I would like to thank my Graduate Project Supervisor Dr. Zehai Zhou and committee members Dr. Omar El-Gayar and Michael Traina for their support and life-long dedication towards education. Without their guidance, expertise and thoughtfulness I would not have been able to complete such a daunting task. I am, also, indebted to all the wonderful instructors at DSU whom I have had the pleasure of knowing and working with throughout my Graduate studies. They have motivated me to set and attain a higher level of goals I would not have done otherwise.

I would also like to thank ITT Norwood for providing me the opportunity to create a “real-world” application for my thesis project. I want to extend a special “thank you” to Carolyn Otteson for all her hard work creating the paper-based tutor system currently being employed at ITT Norwood and having the kindness to share all her carefully-crafted information with me. Lastly, I want to extend a word of thanks to my fellow staff members that contributed to the success of the project, in many ways, by providing either technical or background support.

Finally, I would like to extend some personal thanks. First and foremost, I want to thank my Lord, Jesus Christ, for the patience and understanding that through him all things are possible. Secondly, I want to thank my wife, Karen Cheser, for all her support. Her patience, love and understanding throughout the course of my time at DSU has reminded me over and over again the reasons why I married her. Finally, I would be remiss if I did not

mention my two boys, Will Henry and Wyatt; it is because of them that I spend so much time and energy bettering myself. It is my hope that all the hard work I endure today will provide them a bright and prosperous future.

TABLE OF CONTENTS

1. INTRODUCTION	1
1.1 ITT Educational Services, Inc.....	1
1.1.1 Who They Are.....	1
1.1.2 History.....	1
1.1.3 Programs of Study.....	2
1.1.4 Mission Statement.....	3
1.1.5 Future Plans for Growth.....	3
1.2 What is Tutoring?	4
1.2.1 The Benefits of Tutoring.....	4
1.2.2 Why Tutoring Works	5
1.2.3 Adult Learning Theory	6
2. PROBLEM DEFINITION AND DESCRIPTION	8
2.1 Overview of the Current Situation.....	8
2.2 Enrollment and Selection of Tutors	8
2.3 Identification of At-Risk Students	9
2.4 Tutoring Sessions.....	9
2.5 Tracking of Tutoring.....	9
2.6 ITT Demographics	10
2.7 The Problem.....	10
3. PROJECT OBJECTIVES AND DELIVERABLES.....	12
3.1 The Objectives	12
3.2 The Deliverables	16
3.3 Desired Outcome	17
4. SCOPE OF THE PROJECT	18
4.1 Analysis.....	18
4.2 Discovery	18
4.3 Technology Analysis	21
4.4 Use Case Analysis.....	25
4.5 Information Architecture	27
4.6 Prototyping.....	28
4.7 Database Development	30
4.8 Web Page Development.....	33
4.9 Administrative Development.....	33
5. RESULTS AND CONCLUSIONS.....	41
5.1 Objectives Review	41
5.2 Lessons Learned.....	41
5.3 Additional Recommendations and Considerations.....	43

REFERENCES	44
APPENDIX A: SYSTEM REQUEST	46
APPENDIX B: INTERVIEW REPORT	48
APPENDIX C: USE CASES	51
APPENDIX D: SQL STATEMENTS	60
APPENDIX E: SITE MAP	68
APPENDIX F: SCREEN SHOTS AND SOURCE CODE	69

TABLE OF FIGURES

Figure 1: Northern Kentucky University TutorTrac	19
Figure 2: Use Case Diagram	26
Figure 3: Information Architecture	27
Figure 4: Template Design – index.dwt.....	29
Figure 5: ITT Tutor System E-R Diagram.....	30
Figure 6.1: Admin Control Panel.....	34
Figure 6.2: Retrieve User Page	35
Figure 6.3: Update User Search Page	36
Figure 6.4: Update User Page	37
Figure 6.5: Delete User Page	38
Figure 6.6: View Sessions Page.....	39
Figure 6.7: View Ratings Page	40

1. INTRODUCTION

This report details the work completed for an internal tutoring system for ITT Technical Institute, Norwood, Ohio. The system is a web-based local portal for students, tutors, faculty, staff and administration. It will serve as a centralized point for communication and tutor session recording. This section will seek to provide background information on the following topics: ITT Technical Institute Norwood's parent company ITT Educational Services (ITT/ESI), tutoring, and adult learning theory. Before such an endeavor can be undertaken it is essential that the background history and direction of ITT/ESI be understood, as well as, a more thorough understanding of what tutoring means and how adults best learn.

1.1 ITT Educational Services, Inc.

1.1.1 Who They Are

ITT Educational Services, Inc. is the leading provider of technology-oriented postsecondary degree programs in the United States based on revenue and student enrollment. The company offers associate, bachelor and master's degree programs and non-degree diploma programs to approximately 38,000 students at 77 ITT Technical Institutes located in 30 states.

1.1.2 History

ITT/ESI is a Delaware corporation incorporated in 1946. From 1966 until the initial public offering on December 27, 1994, ITT/ESI was a wholly-owned subsidiary of ITT Corporation. Following the IPO, ITT held approximately 83% of ITT/ESI's common stock.

In February 1998, Starwood Hotels & Resorts Worldwide, Inc. acquired ITT. Public offerings of ITT/ESI common stock by ITT—in June 1998 and February 1999-- and ITT/ESI's repurchase of 1.5 million shares of ITT/ESI common stock from ITT (in February 1999) completely eliminated ITT's beneficial ownership (FAQ).

1.1.3 Programs of Study

Curriculum offerings, leading primarily to associate's and bachelor's degrees, are designed to help students begin to prepare for career opportunities in various fields involving technology, including: electronics, computer network systems, computer drafting and design, web development, multimedia, software applications and programming, industrial design, automated manufacturing, computer visualization, telecommunications and other areas. Programs of study vary among the ITT Technical Institutes.

Most ITT Technical Institute programs of study blend traditional academic content with applied learning concepts, with a significant portion devoted to practical study in a lab environment. Advisory committees, comprised of representatives of local businesses and employers, help each ITT Technical Institute periodically assess and update curricula, equipment and laboratory design.

In 2001, ITT Tech began offering online education. Programs leading to a Bachelor of Science degree are currently available in Information Systems Security and Technical Project Management for Electronic Commerce. Additionally, ITT Tech offers an online MBA program (About Us).

1.1.4 Mission Statement

The mission of ITT Educational Services, Inc. and ITT Technical Institutes is to provide a quality postsecondary education and the services that can help a diverse student body prepare for career opportunities in various fields involving technology.

ITT Educational Services, Inc. and ITT Technical Institutes strive to establish an environment for students and employees, which promotes professional growth, encourages each person to achieve his or her highest potential and fosters ethical responsibility and individual creativity within a framework of equal opportunity (FAQ).

1.1.5 Future Plans for Growth

ITT/ESI developed a 10-Point Enrollment Growth Plan. The elements of the Plan are as follows:

1. Increase enrollment in existing programs offered at existing colleges.
2. Open new college locations.
3. Open learning sites to supplement enrollment growth at existing locations.
4. Raise the program offerings to the bachelor degree level at additional colleges.
5. Develop and offer new degree programs in various fields of technology.
6. Research, develop and offer non-technology degree programs.
7. Increase the number of degree programs offered entirely online
8. Expand the use of the 2+1 hybrid delivery model.
9. Seek international growth opportunities.
10. Evaluate the potential of offering non-degree programs of study.

(FAQ)

1.2 What is Tutoring?

Peer tutoring is a very old practice, traceable back at least as far as the ancient Greeks.

Archaic definitions of peer tutoring perceived the peer tutor as a surrogate teacher, in a linear model of the transmission of knowledge, from teacher to tutor to tutee. Later, it was realized that the peer tutoring interaction was qualitatively different from that between a teacher and a student, and involved different advantages and disadvantages. (Topping).

1.2.1 The Benefits of Tutoring

Tutoring can have benefits for both the tutor and the tutee. The following serves to detail some of the many benefits of tutoring.

Benefits to the tutor:

- Encourages life-long learning.
- Develops critical thinking skills.
- Provides material review.
- Builds confidence.
- Increases depth of understanding of subject area.
- Provides structured learning and study strategies.
- Increases overall subject and general knowledge.
- Improves attitude toward academic achievement.
- Establishes empathy and connectedness with peers.

Benefits to the tutee:

- Motivates the student to succeed
- Provides individual structured learning.
- Provides guided practice outside the classroom.
- Improves academic performance and personal growth.
- Increases attitude toward learning.
- Generates stronger effects than other individualized teaching strategies.
- Instills self-directed learning.
- Improves self image.

1.2.2 Why Tutoring Works

The primary reason peer tutoring works so well is that the tutor speaks a language similar to the tutee's. Often the teacher can overwhelm a student with terminology that the student does not understand. That same student may be hesitant to ask for clarification for fear of appearing ignorant. The closeness in age and experience between the peer tutor and tutee places the tutee in a less passive role. This, in turn, can create more interaction and involvement in the learning process resulting in greater retention. Being closer in status to the tutor the tutee can feel freer to express him / her self and risk attempting untested solutions.

1.2.3 Adult Learning Theory

In order to build a successful tutoring system special consideration must be taken to understand how adults learn best. There are special needs and requirements that adults

possess as learners. Malcom Knowles, the pioneer of adult learning, sought to identify these characteristics. Stephen Lieb, a senior technical writer and planner for the Arizona Department of Health Services, during VISION, Fall 1991, described Knowles characteristics as the following:

- Adults are *autonomous* and *self-directed*. They need to be free to direct themselves. Their teachers must actively involve adult participants in the learning process and serve as facilitators for them. Specifically, they must get participants' perspectives about what topics to cover and let them work on projects that reflect their interests. They should allow the participants to assume responsibility for presentations and group leadership. They have to be sure to act as facilitators, guiding participants to their own knowledge rather than supplying them with facts. Finally, they must show participants how the class will help them reach their goals (e.g., via a personal goals sheet).
- Adults have accumulated a foundation of *life experiences* and *knowledge* that may include work-related activities, family responsibilities, and previous education. They need to connect learning to this knowledge/experience base. To help them do so, they should draw out participants' experience and knowledge which is relevant to the topic. They must relate theories and concepts to the participants and recognize the value of experience in learning.
- Adults are *goal-oriented*. Upon enrolling in a course, they usually know what goal they want to attain. They, therefore, appreciate an educational program that is organized and has clearly defined elements. Instructors must show participants how this class will help them attain their goals. This classification of goals and course objectives must be done early in the course.

- Adults are *relevancy-oriented*. They must see a reason for learning something. Learning has to be applicable to their work or other responsibilities to be of value to them. Therefore, instructors must identify objectives for adult participants before the course begins. This means, also, that theories and concepts must be related to a setting familiar to participants. This need can be fulfilled by letting participants choose projects that reflect their own interests.
- Adults are *practical*, focusing on the aspects of a lesson most useful to them in their work. They may not be interested in knowledge for its own sake. Instructors must tell participants explicitly how the lesson will be useful to them on the job.
- As do all learners, adults need to be shown *respect*. Instructors must acknowledge the wealth of experiences that adult participants bring to the classroom. These adults should be treated as equals in experience and knowledge and allowed to voice their opinions freely in class (Lieb).

2. PROBLEM DEFINITION AND DESCRIPTION

2.1 Overview of the Current Situation

The current solution provided at ITT Norwood is a paper-based tutoring system that was created and is headed by a single general education instructor, Ms. Carolyn Otteson. The system is time consuming and its activity is difficult to track. Each quarter, tutors change due to graduation, or new tutors becoming available. Tutoring is accomplished through word of mouth communication. Students in need of a tutor search for the head of the tutoring program, Carolyn Otteson, and she matches them with a student who she believes might want to tutor others.

The Dean of ITT-Norwood, Michael Traina, has provided the following description of the current tutoring program (personal communication, July 20, 2004):

2.2 Enrollment and Selection of Tutors

The school maintains a list of students who have volunteered as tutors for the current quarter. This list of students is obtained through in-class enrollment. During the beginning of each quarter, each instructor polls his or her class for possible tutors. Those students wishing to tutor are added to a list of potential tutors for the quarter. The registrar and the Dean then check the list. Those students who do not meet grade point or attendance criteria are removed from the list. The remaining students are those asked to become tutors.

Additionally, the school registrar prints a report showing high honors students who also demonstrate excellent attendance. The students on this list who are not already tutors are individually asked if they are able to volunteer as a tutor.

Another source of tutors comes from the school Ambassadors. The school has approximately 30 Ambassadors who volunteer in various ways around the school. Part of their opportunity for service is in a tutoring area.

After tutors are selected, each tutor is then asked to submit availability, contact information, and courses or subject areas that they feel comfortable in tutoring.

Finally, the tutor list is displayed on an information bulletin board in the main student lounge. Additionally, the list is copied for each instructor and staff member. In this way, if a student has some academic need, the list of tutors is readily available. In a normal quarter, there are at least 20 – 25 tutors on the tutor list.

2.3 Identification of At-Risk Students

Generally, identification of students in need of tutoring comes from either faculty or staff, or is self-generated by the student. Students who feel the need to have tutoring can check the information board for available tutors and contact them as the need arises. Teachers can also suggest to at-risk students that they seek tutoring.

2.4 Tutoring Sessions

Tutoring sessions occur at agreed-upon times between tutor and tutee. It is the tutee's responsibility to contact the tutor and arrange a mutually convenient time and place.

2.5 Tracking of Tutoring

The tracking of actual tutoring time is a hit-and-miss proposition. The LRC has forms that the tutor completes after a tutoring session, but only in a small percentage of tutoring are these forms completed.

2.6 ITT Demographics

School Census: 580 (approx)

Program census:

Computer Electronics and Engineering Technology 185

Computer Drafting and Design 85

Computer Networking Systems 200

Multimedia 50

Web Development 30

Software Applications and Programming 30

Average Age:

24.5 years

80% Male, 20% Female

Tuition:

Approximately \$375 per credit hour; \$36,000 for two-year degree

2.7 The Problem

The current system is not working due to lack of tutor training and guidance, and students failing to attend scheduled tutoring appointments. Considering there is no overall strategy

for preventing students from dropping out, the school realizes that many of the students' math and composition skills are weak and below state standards. With this in mind, several departments have historically raised concerns about ways to improve student retention. As a result, ITT Norwood has made available many resources, including volunteer student tutors, math-based learning software, instructors who donate their time tutoring, and the ITT Virtual Library. The recruiters use these resources as a marketing tool to help generate student enrollment. However, the tutors are untrained, lack guidance and find it difficult negotiating mutually agreeable times to meet with fellow students. As a result, instructors give up to 12 hours per week of their time tutoring outside the classroom, many of them even coming in on Sundays to accommodate students' schedules. Also, since the current solution has been an ad hoc situation, it was difficult to track tutoring, and student requests for tutors often went unanswered. As a result, at-risk students in need of extra help might not have received the help needed and consequently dropped out of school.

3. PROJECT OBJECTIVES AND DELIVERABLES

After careful evaluation of the current problem Mike Traina and I met to discuss the anticipated objectives and deliverables of the project. Listed below, and explained more fully, are the objectives and deliverables upon which we decided.

3.1 The Objectives

The ITT Tutor System will serve to accomplish the following objectives:

- Secure assistance for the student
- Provide training material for tutors
- Increase communication between the tutor and student
- Provide feedback and progress reports to the students' instructor(s)
- Make available materials to support the classroom experience
- Time and date stamp tutoring sessions
- Reduce the instructor tutor load, so that work time can be devoted to delivering quality instruction and student placement
- Keep an ongoing record of the progress of "at risk" students
- Increase retention

Secure assistance for the student

The first objective was meant to create a system that would be a reference for the student that was in need. The term "diverse" stated within ITT's mission -- *to provide a quality postsecondary education and the services that can help a diverse student body prepare for career opportunities in various fields involving technology* – encompasses a multitude of

individuals. As mentioned earlier, the typical ITT student may be “at risk”—he/ she either did not do well in traditional educational environments or is so far removed from education that they lack the rudimentary skills to become a successful learner. The ITT system would provide a safe and reliable place where this student could seek assistance.

Provide training material for tutors

The ITT Tutor System will serve as a reference source for the tutor. The site will provide various links to sources that will contain training documents, tutoring information and other useful tools. The system will also become a virtual library containing documentation on various topics of interest that relate to tutoring. These documents will be created and converted into Adobe Acrobat files so they can be downloaded and printed on any platform.

Increase communication between the tutor and student

Another objective is that of increased communication. In the past, many students have lost communication with their tutor. This loss of communication has resulted in feelings of hopelessness and has inevitably caused many students to drop out of the college. With the system as intermediary the communication can be tracked and monitored to make sure that this problem does not continue. This heightened communication method will provide students with a more reliable and efficient method of arranging for tutoring.

Provide feedback and progress reports to the students' instructor(s)

Feedback is essential to the success of any system. The system will maintain the current schedule showing open times, tutors, courses, and already scheduled appointments. It will also record session logs showing the time and date of each session, length of session, and a transcript summary of the session. All this data will be used for tracking and reporting purposes. In addition, for historical purposes, the system will provide a contact interface and retain a log of the contacts made.

Make available materials to support the classroom experience

The virtual library component of the ITT Tutor System will also serve the function of providing relevant information to all individuals involved in the tutoring process.

Time and Date Stamp Tutoring Sessions

It has become apparent that for tutoring to be successful sessions must be tracked. Tracking sessions can alert administration to anomalies. For instance, if a particular subject has above average tutoring activity then there may be a need to analyze that course curriculum. Also, if a particular student is constantly seeking help then that would be a red flag to administration that they need to direct their retention efforts towards that individual. Furthermore, tracking sessions can also give some indication as to the efficiency of the system. The more activity generated would equate to a successful embracing of the system by students, staff and faculty.

Time and date stamping will also provide a chronology of sessions that will provide administration with the ability to award tutors based on hours served. The total number of hours will not go unnoticed, as is often the case with a paper-based system, and be used to appropriately reward the tutor.

Reduce the instructor load, so that work time can be devoted to delivering quality instruction and student placement.

With a system taking the brunt of the work from the instructor, the instructor is freed to devote more time and energy to their individual students. With the instructor spending more time on curriculum enhancement and teaching methodology the overall quality of instruction will increase. This again can be directly linked to increased retention.

Keep an ongoing record of the progress of "at risk" students

The data points acquired by the system will help administration identify those students who are consistently seeking help. Armed with this information they can then focus their efforts towards providing more ancillary support for these particular individuals and hopefully increase retention.

Increase Retention

ITT Technical Institute Norwood needs a proactive and coordinated plan for retaining its students. The current dropout rate is 5-7 students per quarter and, though there are many reasons why students drop out, one factor that has been identified as a problem is the availability of tutoring. Students facing difficult classes are not always able to obtain

assistance. In the current system, finding tutors and scheduling students to meet with them is done by word-of-mouth. In addition, there is no formalized training for tutors, the need for and use of tutors is not currently tracked, and faculty are often pressed to work additional hours to fill in for this need. Most student drops occur in the first and second quarter, when tutoring is the most important. With a better tutoring management system in place, it is possible to save students from dropping (See Table 1).

Table 1: Students per quarter saved from dropping

# Students per quarter saved from dropping	Yearly	Total Saved Tuition
1	4	\$144,000
2	8	\$288,000
3	12	\$432,000

(source: M. Traina)

3.2 The Deliverables

The deliverables for the ITT tutoring system include:

- (19) HTML pages
- (37) PHP pages
- (1) MySQL database containing the following 8 tables:
 - Admin
 - Course
 - Courses_Taught
 - Login

- Rate_Tutor
 - Student_Information
 - Teachable_Courses
 - Tutor_Session
- Documentation

3.3 Desired Outcome

At the completion of the project ITT will have a “one stop” tutoring system that will provide the following functionalities. The system will give tutees available links and reference information about tutoring, search for available times, courses, and tutors, and rate tutoring sessions. Tutors will be able to log tutoring sessions, communicate with students, and access training materials. This system will serve to intentionally weave the various current resources together to form a safety net for ITT’s students. If the “at risk” student needed help that is beyond the scope of the instructor and if it were appropriate to do so, the instructor will have the capability to refer the student to a localized equipped system.

4. SCOPE OF THE PROJECT

This chapter describes the ITT Tutoring System project from a process perspective. It should serve to describe the process that was followed to complete the system for ITT Norwood, Ohio.

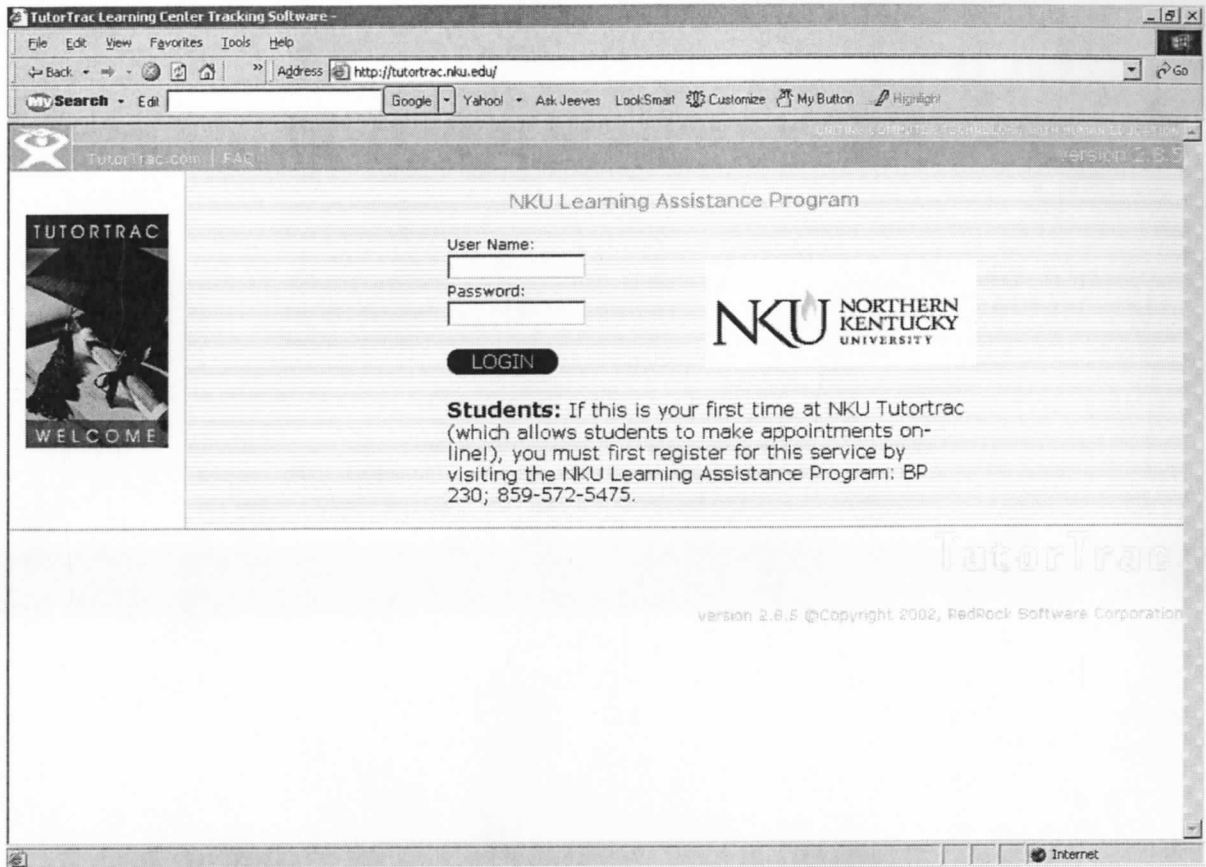
4.1 Analysis

The task of designing and developing a web-based tutoring system for ITT was a daunting one. There were many steps involved in the process and many unexpected hurdles to cross. My first step was to gather all relevant tutoring materials currently developed. I found that the current tutoring system was nearly non-existent. The existing paper-based system was loosely constructed and not fully implemented. And even though I had been with ITT for 3 years I was unfamiliar with the current tutoring system. Therefore, my initial primary goal was to solicit information from the Dean and try to understand what the intended system needed to accomplish. To accomplish this task I chose to conduct an interview (See Appendix B).

4.2 Discovery

Once I gathered as much information locally as I could, I began an Internet search on similar systems. The system that I used as a reference point was the tutoring system provided by a local state university – Northern Kentucky University – called TutorTrac, see Figure 1. I had a pseudo login provided by a local faculty member that allowed me to navigate through the system as an observer. It was through this exploration that I gathered the needed information to begin structuring the system I would create for ITT.

Figure 1: Northern Kentucky University TutorTrac



Once I delved deeper into NKU's system I realized that I was navigating through a proprietary solution. It became apparent that TutorTrac was a software solution provided by a company called Redrock Software Corporation. And although TutorTrac appeared to have all the functionality that ITT desired, the cost of the software (\$2,495) well exceeded the intended budget for this project (\$0). The following excerpt from Redrock Software's website better describes their product and will paint a clearer picture of the type of system ITT administration envisioned.

TutorTrac is the ultimate management software for learning, writing, reading, tutoring departments, and academic skills centers for traditional and athletic students.

Perhaps the most unique feature of TutorTrac is that it is web-based. Tutors, students and administrators can access their records via the web. Our system is secure, so unauthorized access is not even a possibility. This gives every one using the system the ability to get reports, demographics, make requests, manage scheduling and manage center resource materials at their own convenience. Users cannot access the login/out features via the web, so usage reports remain accurate.

TutorTrac is an easy to use web-based tutor and learning center management tool!

TutorTrac is able to store over 2 billion student records. This makes it viable for large institutions. And importantly, whether your center has 200 or 2 million records, searches occur accurately and quickly. Centers can import their data directly from other products, systems or formats they are using. Centers can access both current and past data. Users can instantly view students logged into the system. An optional bar-code reader or card scanner can be utilized for logging in and out.

Our reporting feature offers standardized reports and the ability to customize the data you need. We also have a built-in messaging system that allows administrators and tutors to send a message to one or all students. TutorTrac can also manage seminars and workshops with ease.

Plus, TutorTrac can send Email notification to tutors and students reminding them about upcoming appointments automatically!

(Tutor and Learning Center Management Tracking Software)

4.3 Technology Analysis

Before jumping into the deep end, so to speak, I had to determine the underlying technologies I would use to create and support the new system. I would need technologies for editing graphics, HTML layout, server-side scripting, database design / development and web serving.

Graphics Editing

As a long time designer I have always had my preference of design tools. I have been using Adobe products for more than 14 years and have been hard-pressed to find any company that can compete with them. Adobe's flagship product, Photoshop, is by far the best solution on the market for creating robust web-based imagery. Photoshop's Save for Web option would allow me the ability to control the compression rate of my images without sacrificing quality. Even though the system would be internal I was still concerned about creating small byte count imagery. If this system was ever going to make the leap towards the Internet it was critical that the pages not be over 30KB in size, in order for download time to remain under 10 seconds across a standard dial-up.

HTML Layout

Even though Adobe has a corner on the design market, I will have to say Macromedia has a distinct advantage within the web market. It was with very little deliberation or hesitation that I chose Dreamweaver as my WYSIWYG (What You See Is What You Get) editor. I have been using Dreamweaver for quite some time and am very familiar with the interface. I am particularly fond of Dreamweaver's template capabilities. I had decided from the start that I would need to rely heavily on template design for this system since I was unsure of the number of pages to be created. The template option would provide me the ability to create a single file and reuse it throughout the course of my system. This would also give me the ability to spend less time on the front-end and more time on the back. Since the back-end was my point of weakness I needed to embrace this solution if I proposed to come in on time with the system.

Server-Side Scripting

In order to interface with a database and provide a truly dynamic experience I had to choose a server-side scripting language. The two options I had before me were PHP (Hypertext Preprocessor) or ASP (Active Server Pages). Active Server Pages (ASP) is Microsoft's scripting technology for displaying dynamic web pages; ASP programs mostly use VBScript or JScript. PHP is the open source alternative to ASP that runs on multiple operating systems, including Linux, Windows and Solaris. In order to decide which solution would work best I did a bit of Internet research to find the comparison points between the two. As a result, I found that PHP would be a better solution for the following reasons:

1. Speed - PHP is much faster and stable than ASP, ASP is built on a COM-based architecture so when ever a program tries to connect to a database or calls a COM object, there is an overhead on the server. All this COM overhead adds up and slows things down. Under PHP, everything runs in PHP's memory space. This means that PHP code will run faster because there is no overhead of communicating with different COM objects in different processes. Again, ASP is slower and has more memory intensive applications than PHP's model because each ASP language compiler runs in its own process.

2. Price - PHP installations are definitely cheaper to install, PHP runs great on Linux—which is free; on the other hand, ASP runs on Microsoft's IIS Server (Internet Information Server) which need's Windows NT/2000/2003 Servers. Apart from that, ASP primarily uses MS-SQL Server as the back end which is expensive, whereas PHP programmes mostly use MySQL which is again FREE!

3. Cross Platform compatibility - As stated earlier, PHP programs run on Unix, Linux, Solaris and Windows. With ASP the user must primarily use Windows.

(PHP vs. ASP)

Database Design / Development

When it came time to choose a database I again went with the open source solution. The following quote from MySQL.com gives sufficient reasons for my decision.

The MySQL database server is the world's most popular open source database. With more than five million active installations, MySQL has quickly become the core of many high-volume, business-critical applications.

Customers such as Yahoo!, Google, Cisco, Sabre Holdings, HP and NASA are realizing significant cost savings by using MySQL's high performance, reliable database management software to power large Web sites, business-critical enterprise applications and packaged software applications (MySQL).

Also in terms of pure performance, MySQL is the leader, primarily due to its default table format, MyISAM. MyISAM databases are very compact on disk and place little demand on CPU cycles and memory.

Web Serving

Considering PHP was my server-side scripting language of choice it was only natural to chose Apache as my web server. Besides the interoperability of PHP with Apache I found through my research that Apache had many benefits over the Microsoft's IIS. Ryan Bloom, manager of core engineering, Covalent Technologies, has been a member of the Apache Software Foundation since 1999. Bloom has played a leading role in the development of Apache 2.0, which became available in April. Bloom is also the author of Apache Server 2.0: The Complete Reference. He states in the **June 18, 2002 SearchWin2000 article, *Kicking the Windows Habit: Apache vs. IIS***, the advantages of using Apache Web Server:

The most important advantage Apache had over IIS was its parent and child process model. Apache doesn't serve requests. Its only job is to make sure there is one child to serve a

request. So if something happens to the one child process serving the request, another child process is brought up as quickly as possible. Previous to IIS 6.0 there was no multi-process model. If the Web server died, you didn't get it back until you started it up again. When you are doing complex stuff like Web serving, you take the chance that the process serving the request will fail. With IIS, if that third party code falls, that's it. It's done. With Apache, we just recreate the process. IIS 6.0 has gone to this model but it is moving a lot of the Web server down into the kernel. I want the kernel to be small and tight with as little code as possible. The more you put in the kernel the greater chance your operating system will fail. There are companies that have created Web servers for Unix that are in the kernel, but they are optional.

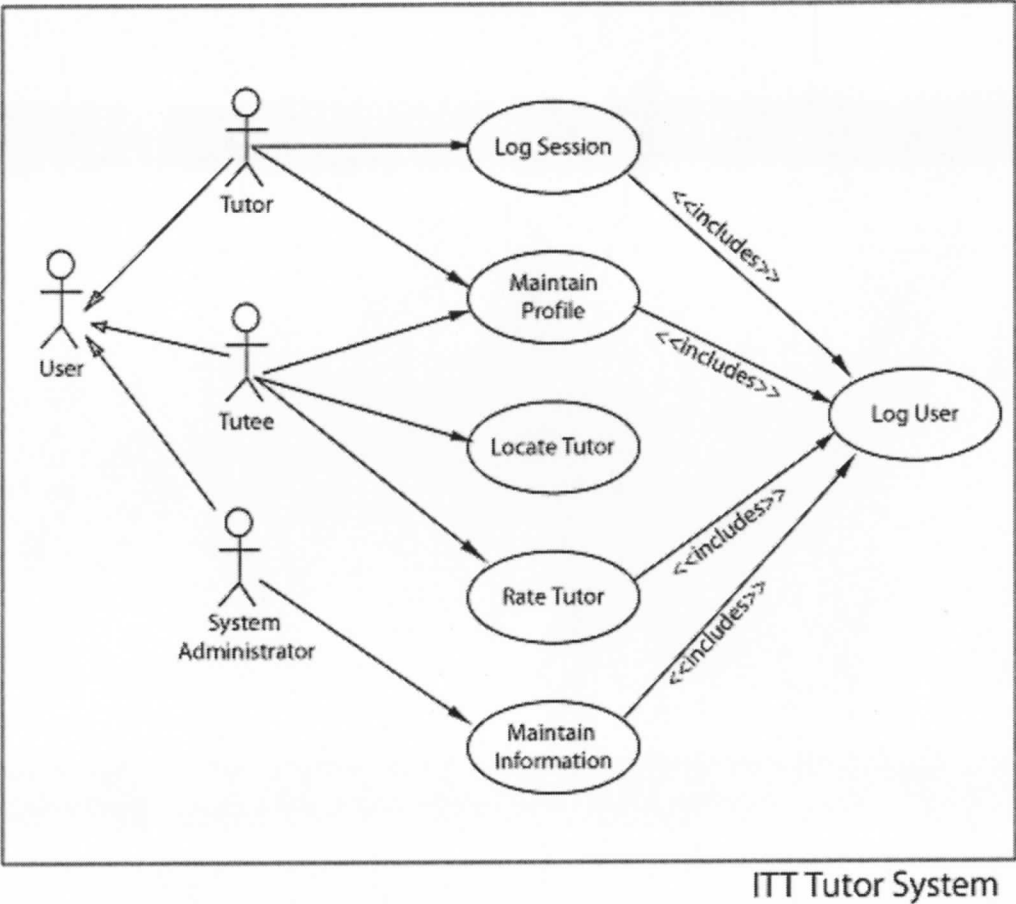
Another benefit of Apache over IIS is that it is easy to extend. There is a whole community that is built up around Apache so if you need to do something that isn't out of the box, chances are you can find someone who has done it and you can download their model. And finally, security is a reality of life. There continue to be viruses and worms that target IIS. Apache issues security warnings occasionally, but they tend to be limited in scope. If there is a security hole in Apache, often you can work around it with a code fix, or you can change your configuration to work around the problem, depending on what the problem is, of course (Bloom).

4.4 Use Case Analysis

The text, *Systems Analysis and Design, An Object-Oriented Approach with UML*, defines a use case as a formal way of representing how a business system interacts with its

environment. A use case illustrates the activities that are performed by the users of the system. Before I started the development of the system I needed to understand how the various users of the system would interact with it. Understanding this step would help me better design the navigation and develop my SQL statement code. Figure 2 shows the use case diagram. For more information on use cases see Appendix D. (Dennis 152).

Figure 2: Use Case Diagram



4.5 Information Architecture

Information Architecture, as defined by **TechDis.com**, is the organization of a website's structure and content, the labeling and categorizing of information and the design of navigation and search systems. *Figure 3* is an overview of the information architecture for the ITT Tutor System.

Figure 3: Information Architecture

Templates	All .dwt templates reside in this location
Admin	Administrator side information
<i>Add_user</i>	Script to add users
<i>Delete_user</i>	Script to remove users
<i>Ratings</i>	Script to view tutor ratings
<i>Retrieve_user</i>	Script to view users
<i>Sessions</i>	Scripts to view tutor sessions
Application	Contact form to request to be a tutor
Create_profile	Scripts related to adding a profile
<i>Tutor</i>	Script to add a tutor profile
<i>Tutee</i>	Script to add a tutee profile
Edit_profile	Script to edit a profile
Images	All graphics used across the site
Includes	All PHP includes
Locate_tutor	Scripts related to locating tutors
<i>Course</i>	Script to locate tutors by course ID

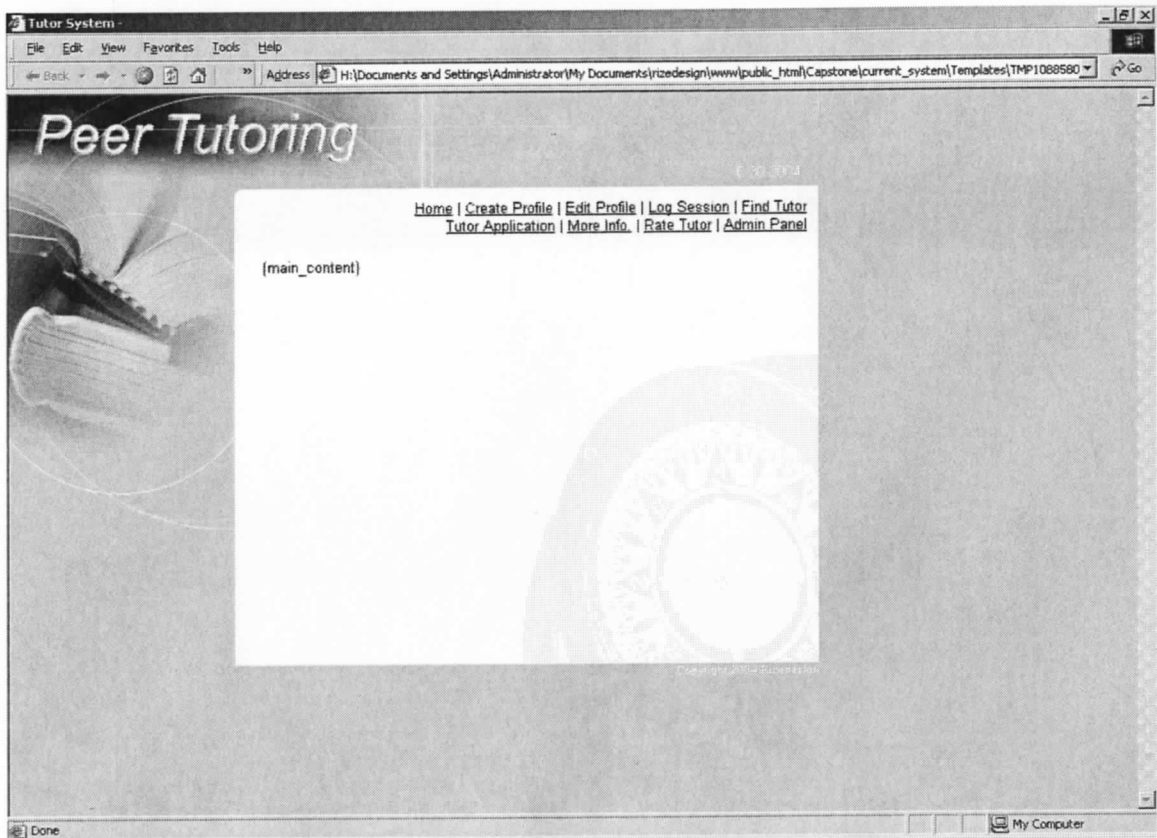
<i>Name</i>	Script to locate tutors by last name
More_info	General information on tutoring
<i>Additional_resources</i>	Links to additional information
<i>FAQS</i>	Facts on tutoring
<i>General_Procedures</i>	Tutoring procedures
<i>How_to_be_a_tutor</i>	Information on becoming a tutor
<i>How_to_make_appt</i>	Information on how to make an appointment
<i>More_info.</i>	Information on how to use the ITT Tutor System
<i>Recognizing_tutors</i>	Information on the rewards of tutoring
<i>Tips_for_tutors</i>	Helpful tips on tutoring
<i>Tutee_responsibilities</i>	Responsibilities of tutees
<i>Tutor_responsibilities</i>	Responsibilities of tutors
<i>What_tutoring_can_do</i>	Information on the benefits of tutoring
<i>Why_tutor</i>	Reasons to tutor
Ratings	Script to rate tutors
Session	Script to log tutoring sessions

4.6 Prototyping

After the objectives, information architecture and use cases were well defined, I began creating visual mock-ups of the graphical user interface (GUI). I completed this process with pen and paper. Coordinating with the Dean on color choices and wording, I was able to narrow my choice to a single design. After only one prototype design both the Dean and I were happy with the result. Luckily, I did not have to go through the iteration process more

than once, allowing me to focus more time and energy on the back-end. I then translated the hand-drawn design into HTML. Once I had the design coded to my liking I used Dreamweaver's, Save as Template option, to create my .DWT file for the remainder of the system. *Figure 4* displays a screenshot of the template design.

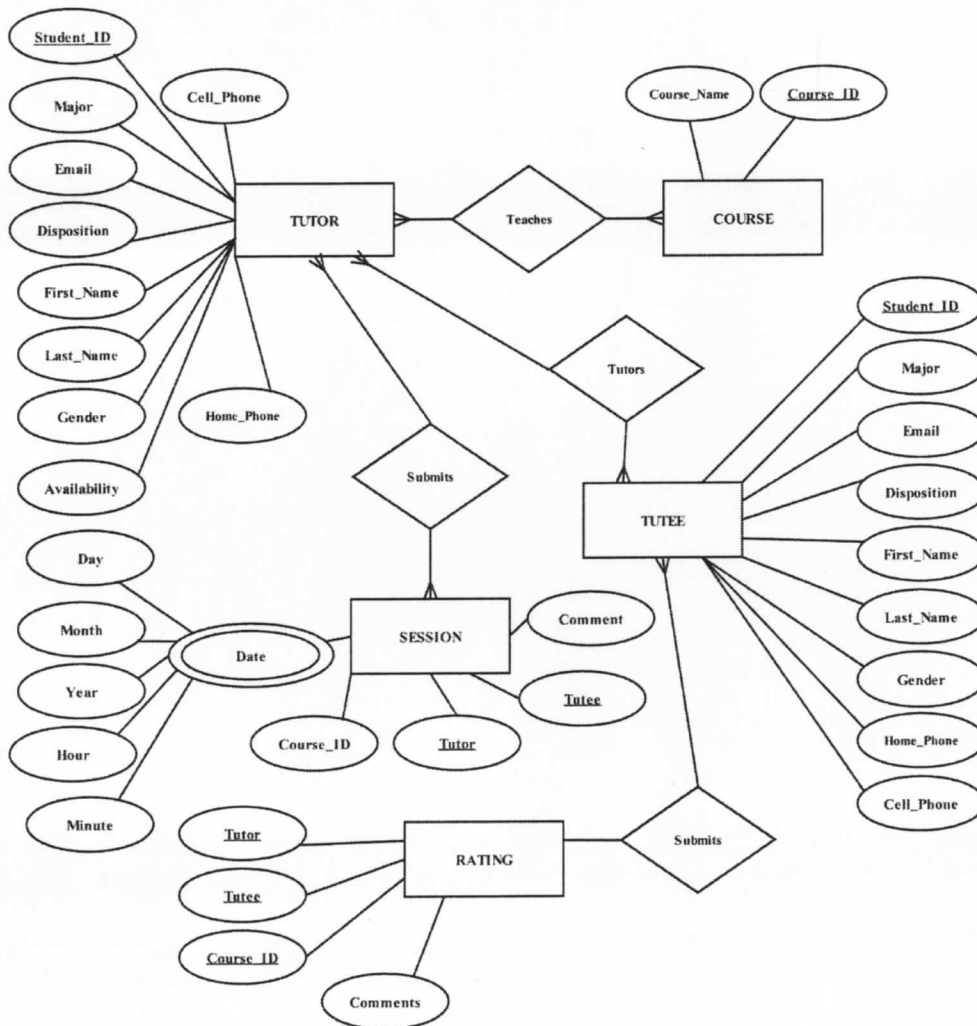
Figure 4: Template Design – index.dwt



4.7 Database Development

Following a meeting with the Dean—during which we listed the required entities and attributes of the system—I created the entity-relationship diagram of the ITT Tutor System, represented in *Figure 5*.

Figure 5: ITT Tutor System E-R Diagram



When I completed the logical design of the ITT Tutor System database I ran PHPMyAdmin and began constructing my tables. See Table 2 for the ITT Tutor System Database Schema.

This led to the next logical step of trying to figure out what SQL statements I needed to develop in order to interface with the database. Utilizing my use cases as a guide, I was able to readily write queries. I decided to write the queries on paper first before coding, in order to make sure I had it right. I would not translate my pseudo code into actual workable code until after I completed the HTML pages. (See Appendix D for a listing of all SQL statements used in the ITT Tutor System.)

Table 2: ITT Tutor System Database Schema

	Type	Null
[Admin]		
<u>Username</u>	varchar(50)	No
<u>Password</u>	varchar(50)	No
[Course]		
<u>Course_ID</u>	varchar(20)	No
Course_Name	varchar(50)	No
[Courses_Taught]		
<u>Student_ID</u>	varchar(8)	No
<u>Course_ID</u>	varchar(8)	No
Comments	varchar(200)	No
[Login]		
<u>Student_ID</u>	varchar(8)	No
Username	varchar(8)	No
Password	varchar(8)	No

[Rate_Tutor]		
<u>Tutor ID</u>	varchar(50)	No
<u>Tutee ID</u>	varchar(50)	No
<u>Course</u>	varchar(50)	No
Comments	varchar(200)	No
[Student_Information]		
<u>Student_ID</u>	varchar(8)	No
First_Name	varchar(20)	No
Last_Name	varchar(20)	No
Email	varchar(20)	No
Home_Phone	int(13)	No
Cell_Phone	int(13)	No
Gender	varchar(1)	No
Major	varchar(20)	No
Disposition	varchar(1)	No
Availability	varchar(100)	No
[Teachable_Courses]		
<u>Student_ID</u>	varchar(20)	No
Course_Name	varchar(20)	No
[Tutor_Session]		
<u>Tutee ID</u>	varchar(20)	No
<u>Tutor ID</u>	varchar(20)	No

Comments	varchar(200)	No
Course	varchar(20)	No
Day	char(2)	No
Month	varchar(15)	No
Year	varchar(4)	No
Hour	char(2)	No
Minute	char(2)	No

4.8 Web Page Development

Creating the HTML pages was the next logical step in the process. With all the objectives and goals defined, the template created and the database built, it was time to knock-out the pages of the site. I began by writing the content for each page in a blank HTML file. I then attached each page to the index.dwt template file, creating uniformity across the entire site. After getting the basic pages laid-out it was time to write the PHP pages and add interactivity via SQL statements.

4.9 Administrative Development

If this system was going to be useful for administration then I needed to build an admin component. This secure access point would allow non-technical administrators the ability to perform various functions based upon the data stored in the ITT Tutor System database. The various types of functions they could perform would include viewing users, adding users,

deleting users, viewing of logged tutoring sessions and viewing of tutor ratings. See Figures 6.1 through 6.7 for screenshots of the Administration site.

Figure 6.1: Admin Control Panel

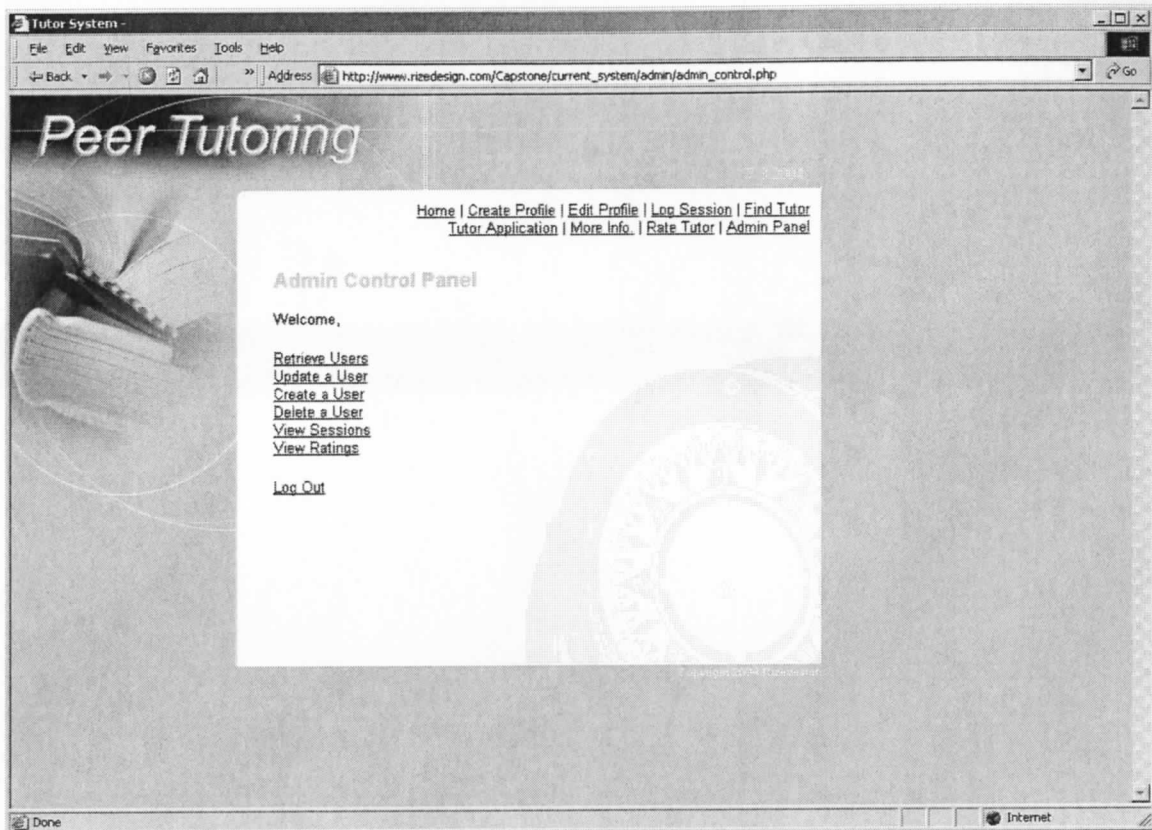


Figure 6.2: Retrieve User Page

The screenshot shows a web browser window titled "Tutor System". The address bar contains the URL: http://www.rizedesign.com/Capstone/current_system/admin/retrieve_user/retrieve_user.php. The page content includes a navigation menu with links: [Home](#), [Create Profile](#), [Edit Profile](#), [Log Session](#), [Find Tutor](#), [Tutor Application](#), [More Info](#), [Rate Tutor](#), and [Admin Panel](#). Below the menu, it states "There are 4 records." and displays a table with the following data:

Student ID	Last Name	First Name	Email	Home Phone	Cell Phone	Major	Gender
11111112	Bonilla	Luis		513-661-5112	D	CEET	male
11111113	Cole	Adam		856-461-3057	D	CEET	male
11111114	Orause	Edwin		513-290-1052	D	CAD	male

Below the table, there is a [Home](#) link. The browser's status bar at the bottom shows "Done" and "Internet".

Figure 6.3: Update User Search Page

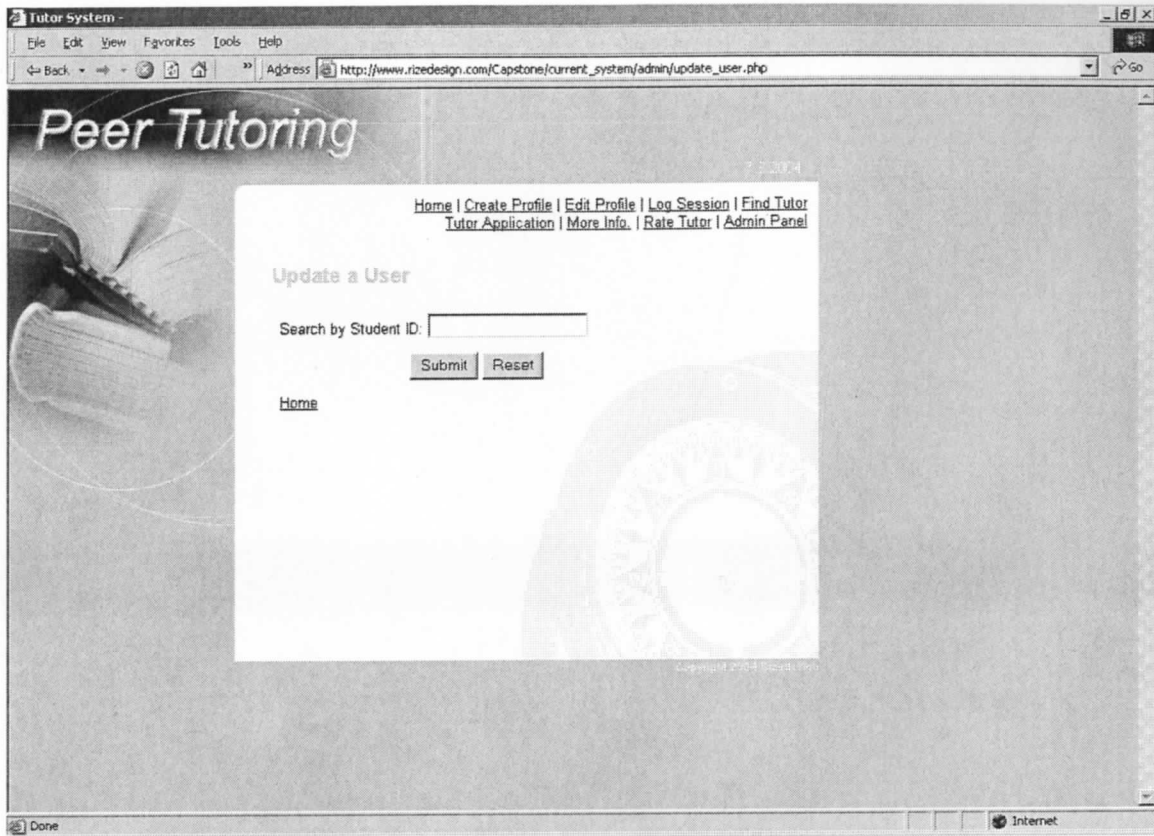


Figure 6.4: Update User Page

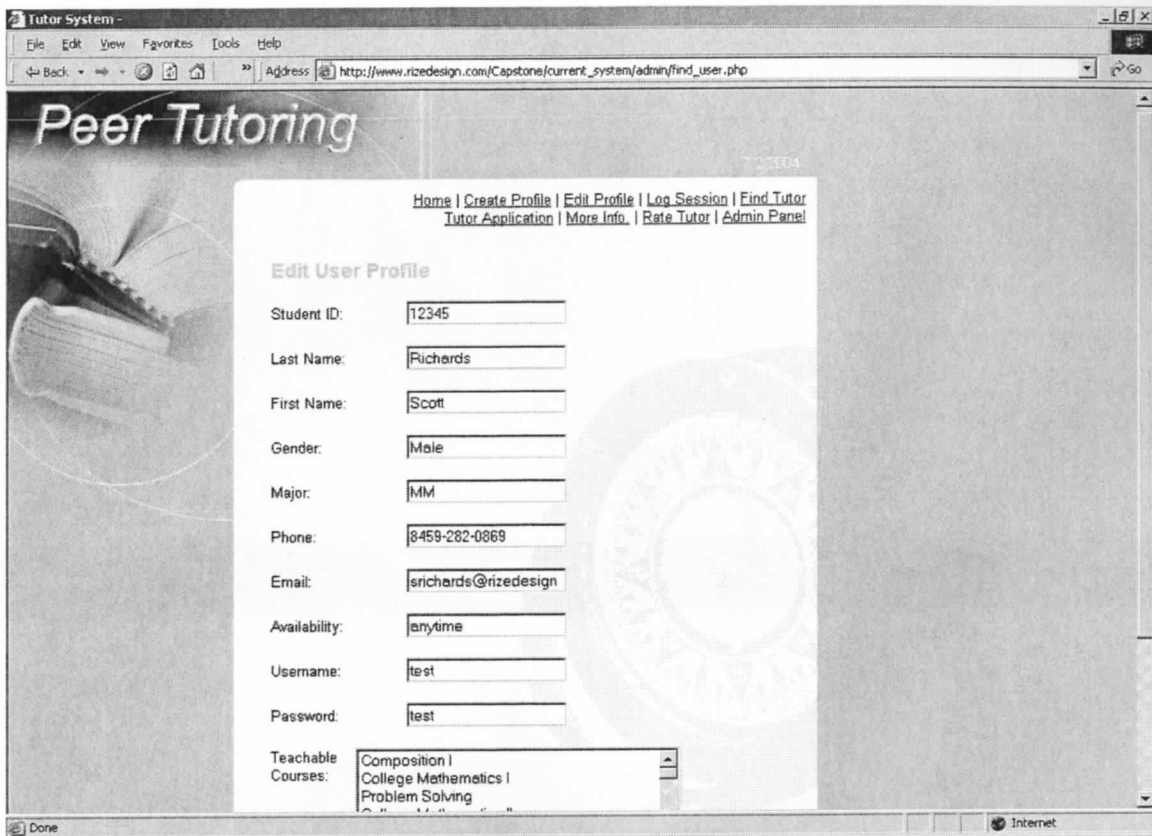


Figure 6.5: Delete User Page

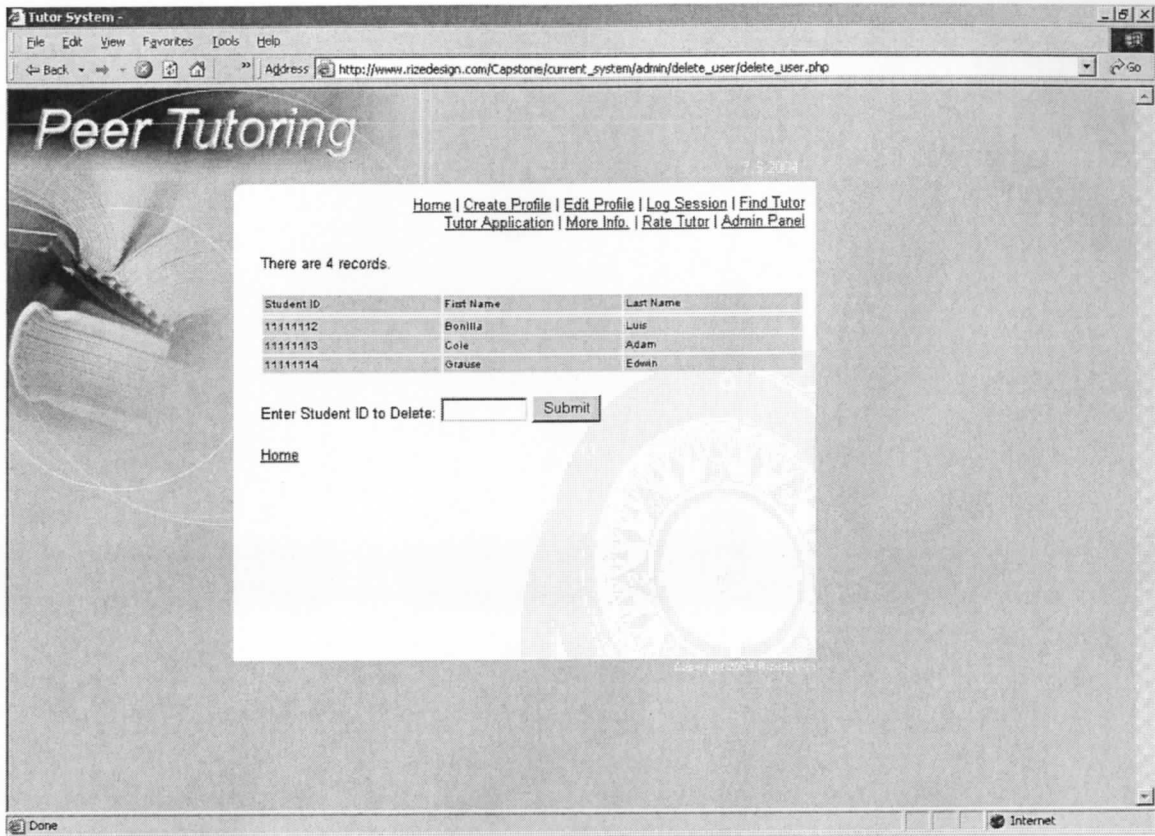


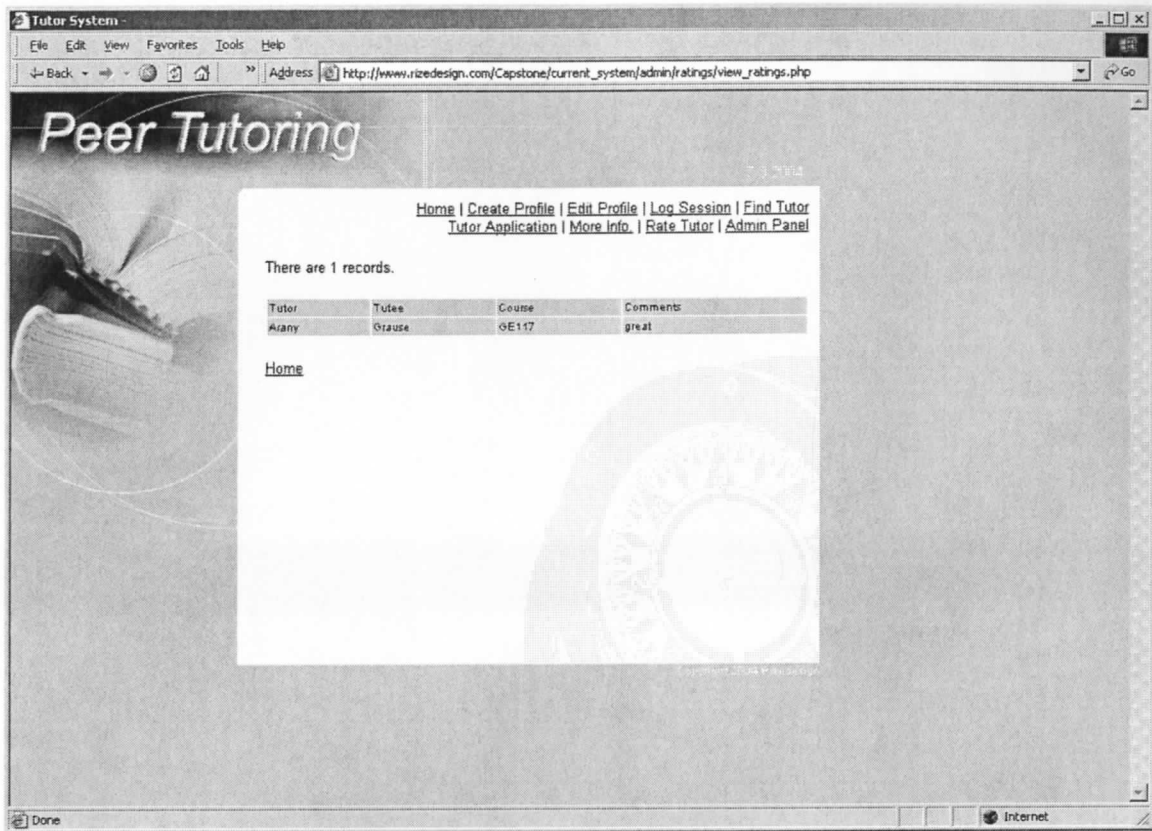
Figure 6.6: View Sessions Page

The screenshot shows a web browser window titled "Tutor System". The address bar contains the URL: http://www.rizedesign.com/Capstone/current_system/admin/sessions/view_sessions.php. The page content includes a navigation menu with links: [Home](#), [Create Profile](#), [Edit Profile](#), [Log Session](#), [Find Tutor](#), [Tutor Application](#), [More Info](#), [Rate Tutor](#), and [Admin Panel](#). Below the menu, it states "There are 3 records." and displays a table of session data.

Tutee	Tutor	Comments	Course	Day	Month	Year	Hour	Minute
			GE117	1	January	2004		
Grasse	mouse		GE117	1	January	2004		
Grasse	Azany		GE117	1	January	2004		

Below the table is a [Home](#) link. The browser's status bar at the bottom shows "Done" and "Internet".

Figure 6.7: View Ratings Page



5. RESULTS AND CONCLUSIONS

5.1 Objectives Review

I did achieve my objective to create a tutoring system for ITT. However, since the system is idle and has not been launched there is uncertainty as to whether or not the system will meet all the objectives outlined in this report. ITT Norwood is awaiting new leadership. In this interim period, there is no one to give the “go ahead” for the tutoring system. As soon as the new Dean is appointed, the system should be implemented. The anticipated deliverable, namely the ITT system, is complete and intact. I have the system stored in three locations for the sake of redundancy.

5.2 Lessons Learned

This project taught me a lot about systems thinking and design. Previous to this experience, I had only concerned myself with the front-end. Now I have become more holistic in thought and am confident I can handle complete systems solutions. I, also, learned a tremendous amount about code. In hindsight, I would have done things much differently. I would have written more include statements rather than rewriting the same code into multiple pages. In addition, I would have relied on W3C standards rather than my old web design ways.

“The World Wide Web Consortium (W3C), along with other groups and standards bodies, has established technologies for creating and interpreting web-based content. These technologies, which we call ‘web standards’, are carefully designed to deliver the greatest benefits to the greatest number of web users while ensuring the long-term viability of any document published on the Web. Designing and building with these standards simplifies and

lowers the cost of production, while delivering sites that are accessible to more people and more types of Internet devices. Sites developed along these lines will continue to function correctly as traditional desktop browsers evolve, and as new Internet devices come to market.” (<http://webstandardsgroup.org/standards/>)

Site wide look and feel consistency

Tristan Nitot, Standards and Technology Evangelist proposes in *Netscape Communications* (March 2003) that by separating structure (or content) from presentation, web designers have a lot to gain. Specifically, presentation is defined using layout-oriented CSS language. Storing CSS in a separate document (aka style sheet) and applying it to a set of HTML documents permits a complete change of presentation for all these documents in a snap. Strict HTML (as opposed to the often-used transitional HTML markup), forces the designer not to use presentation tags in the HTML documents, naturally enforcing the separation of content from presentation.

Improved User Experience: Uses less bandwidth, loads faster

Furthermore, Nitot explains that HTML code is often much more compact (therefore easier to read and maintain) when used in conjunction with CSS. According to various reports, case studies, and Andy King, author of *Speed Up Your Web Site: Web Site Optimization*, CSS has made it possible to transform table-based layouts into CSS-based layouts. Typically this reduces page size by 25 to 50%. This translates into better user experience, according to Usability Guru Jakob Nielsen, who notes that users tend to close a web page when it takes more than 10 seconds to load.

5.3 Additional Recommendations and Considerations

In order for this system to “be all it can be”, I envision a document mill integration. The system could be a “one-stop shop” for every aspect of ITT’s tutoring needs. This would include the ability to populate template certificates and letters of recommendation with system information (e.g. names, hours tutored, etc.) and print directly to our color laser printer, or perhaps our Xerox document center.

Another future add-on that would take the system to the next level of interaction would be an instant messaging component. Rather than have the tutee take down tutor information and proceed to make contact, we could have the system do it for them. Beyond instant messaging, a possible bulletin board or discussion forum would be a nice addition. The more the system can accomplish for the student the more successful it will be.

Apart from the above-mentioned enhancements there are some minor design changes I would implement. For instance, I would probably change all text-based links to graphic images. I would also include more imagery within the site. Possible examples may include pictures of tutors, tutees and shots of the Learning Resource Center (location of tutoring).

REFERENCES

“About Us.” ITT Technical Institute. 19 Jul 2004 <<http://www.itt-tech.edu/about.cfm>>.

About Web Standards . 2004. Web Standards Group. 1 Aug. 2004

<<http://webstandardsgroup.org/standards/>>.

Bloom, Ryan . “What technical advantages does Apache have over IIS?.” 18 Jun 2002.

Kicking the Windows habit: Apache vs. IIS. SearchWin2000. 19 Jul 2004

<http://searchwin2000.techtarget.com/originalContent/0,289142,sid1_gci833798,00.html>.

Dennis, Alan. Systems Analysis and Design, An Object-Oriented Approach. New York: John Wiley & Sons, Inc., 2002.

Falchikov, Nancy. Learning Together : Peer Tutoring in Higher Education. Routledge, 2001.

“FAQ.” 2002. ITT Technical Institute. 19 Jul 2004

<http://www.ittesi.com/ireye/ir_site.zhtml?ticker=ESI&script=1801>.

“Glossary.” 26 Oct 2002. TechDis: Web Accessibility & Usability Resource. 19 Jul 2004

<www.techdis.ac.uk/seven/glossary.html>.

Lieb, Stephen . 1991. Principles of Adult Learning. VISION. 19 Jul 2004

<<http://honolulu.hawaii.edu/intranet/committees/FacDevCom/guidebk/teachtip/adults-2.htm>>.

“MySQL.” MySQL: The World’s Most Popular Open Source Database. 19 Jul 2004

<<http://www.mysql.com/>>.

“PHP vs. ASP.” 15 Apr 2002. PHPBuddy. 19 Jul 2004

<http://www.phpbuddy.com/sub_articles.php?other_articles=9>.

Topping, K.J. “Effective peer tutoring in further and higher education.” SEDA, Staff and Educational Development Association, Paper 95.

Nitot, Tristan. The Business Benefits of Web Standards. 21 Mar. 2003.

Netscape Communications. 1 Aug. 2004

<<http://devedge.netscape.com/viewsource/2003/why-web-standards/>>.

“Tutor and Learning Center Management Tracking Software.” Redrock Software. 19 Jul

2004 <<http://www.tutortrac.com/Default.html>>.

APPENDIX A: SYSTEM REQUEST

11/14/03

Project Name: ITT Tutor System

Project Sponsor:

Name: Mike Traina, Dean of ITT Technical Institute

Department: IT

Organization: ITT Technical Institute

Phone: 513-531-8300

E-mail: mtraina@itt-tech.edu

Business Need: To increase retention, retain tuition, and increase potential to graduate.

Functionality:

Students will be able to schedule themselves for a tutoring session. Tutors and tutees will be able to message each other. The college will be able to track tutoring and outcomes.

The system will maintain information in these areas:

1. Tutors (Name, program of study, quarter, subject area, availability, contact information)
2. Tutees (Name, program of study, quarter, area of need, contact information)
3. Interface to allow tutees to search for a tutor in a specific need area
4. Interface to permit messages to be saved for a tutor or tutee
5. Data capture and reporting information on tutoring events
6. When did it take place
7. Who was involved

8. How Long
9. What was tutored
10. Outcomes

Expected Value:

Intangible:

1. Enable at-risk students to find help for specific course work
2. Help tutees pass classes, earn higher grades, and possibly stay in school
3. Reduce teachers' workload
4. Allow tutors the chance to cement their learning, and earn notice through letters of recommendation and certificates.
5. Student / instructor satisfaction
6. Increased retention
7. Greater potential to graduate and increase value to work force and community

Organizational Feasibility:

Project Champion(s): Mike Traina, Dean of ITT Technical Institute

Senior Management: Dean of Academic Affairs, ITT Technical Institute

Users: Students at ITT Tech

Other Stakeholders: Faculty, Staff, Administration, Shareholders

Special Issues or Constraints:

1. Each quarter tutors will change due to graduation, or new tutors becoming available.
2. Federal Work-study program – students must qualify
3. System, although web-based, must be closed to the outside world (ITT policy)

APPENDIX B: INTERVIEW REPORT

Person Interviewed: Mike Traina, Dean of ITT Technical Institute

Interviewer: Scott Richards

Date: November 14, 2003

Primary purpose: To gather information regarding the development of an on-line tutoring system to increase retention, retain tuition, and increase potential to graduate from ITT Technical Institute.

Summary of interview: Scott Richards interviewed Mike Traina on November 14, 2003 regarding the establishment of an on-line tutoring system to enhance student retention and improve the tutoring process. The new system will keep track of tutors and tutees, messaging between tutors and tutees, data capture and reporting on tutoring events. The current tutoring system is inefficient and results in higher student drop rates and poor student grades. The new on-line tutoring system could be created using existing hardware and software and maintained through work-study assistants.

Open Items: No open items.

Detailed Notes:

Question 1. What business needs will be met by the online tutoring system?

We'd like the tutoring system to increase retention, retain tuition, and increase potential to graduate by:

- Enabling at-risk students to find help for specific course work

- Helping tutees pass classes, earn higher grades, and possibly stay in school
- Reducing teachers' workload
- Allowing tutors the chance to cement their learning, and earn notice through letters of recommendation and certificates.

Question 2. Give a brief description of how it will work.

The system will maintain information in these areas:

- Tutors (Name, program of study, quarter, subject area, availability, contact information)
- Tutees (Name, program of study, quarter, area of need, contact information)
- Interface to allow tutees to search for a tutor in a specific need area
- Interface to permit messages to be saved for a tutor or tutee
- Data capture and reporting information on tutoring events:
 - When did it take place
 - Who was involved
 - How Long
 - What was tutored
 - Outcomes

Question 3. How is tutoring currently scheduled?

The system is currently done through word of mouth, so if this project doesn't happen it will continue in this fashion. It is currently time consuming and somewhat difficult to track activity. Each quarter tutors will change due to graduation, or new tutors becoming available. Tutoring was done through word of mouth communication. Students looked for the head of

the tutoring program, Carolyn Otteson, and she matched them with a student who she believed might want to tutor. Because this was an ad hoc situation, it was difficult to track tutoring, and student requests for tutors often went unanswered. As a result, at-risk students in need of extra help might not have received the help and often dropped out of school as a result.

Question 4. Explain the Technical Feasibility of the project.

We have in-house work-study students that can keep this maintained. Data will need to be kept up.

Question 5. Explain the Economic Feasibility of the project.

We already have the equipment. Work-study student spends 20 hours per week maintaining web site, gathering updated information and ensuring uptime.

Question 6. How did the idea get started?

The Dean of Academic Affairs suggested the idea.

Question 7. How will it be funded?

Work-study is a federal program – students must qualify.

APPENDIX C: USE CASES

Primary Actors

Tutor: A person who provides tutoring services, accessible from the tutoring website

Tutee: A person who schedules tutoring services through the tutoring website

System Administrator: A person who is responsible for maintaining the ITT Tutoring System

Use Case 1: Maintain Information

CHARACTERISTIC INFORMATION

Brief description

This use case allows the system administrator to create, retrieve, update, and delete (CRUD) tutoring information.

Primary Actor

System administrator

Stakeholders

System Administrator – wants to provide most current information to website from database.

ITT – wants system to be up-to-date and efficient so it provides service to students

Tutors – need to access current information from website

Tutees – want most current information in order to make decisions

Trigger

System Administrator accesses the ITT Tutor System

Preconditions

System Administrator is logged on

Guarantees

Success End Condition

System administrator performs desired maintenance operations and any changes in the data are saved

Failed End Condition

Nothing happens

MAIN SUCCESS SCENARIO

1. ITT Tutor System presents the system administrator with maintenance options
- 2a. System administrator request to 'Add User'
3. System administrator submits updated information.
4. ITT Tutor System validates information according to information rules.
- 5a. ITT Tutor System saves tutoring information and presents the system administrator with a confirmation.

EXTENSIONS

- *a. System administrator decides to quit:
 - *a1. ITT Tutor System asks the system administrator to save changes (if any):
 - *a1a. System administrator chooses to save:
ITT Tutor System saves changes and exits [success].
 - *a1b. System administrator chooses to quit:
ITT Tutor System discards any intermediate data and exits [fail].
- 2b. System administrator request to 'Retrieve User':
 - 2b1. System administrator submits search request:
 - 2b2. ITT Tutor System searches for tutor information:
 - 2b1a. Tutor information not found:
ITT Tutor System notifies system administrator [repeat].
 - 2b1b. Tutor Information found:

- ITT Tutor System presents information.
- 2c. System administrator request to 'Update User':
 - 2c1. System administrator submits search request:
 - 2c2. ITT Tutor System searches for tutor information:
 - 2c1a. Tutor information not found:

ITT Tutor System notifies system administrator [repeat].
 - 2c1b. Tutor information found:

ITT Tutor System presents information:

System administrator provides updated information:

ITT Tutor System validates tutoring information according to information rules.

Invalid information:

ITT Tutor System notifies system administrator [repeat].

ITT Tutor System saves information and presents the system administrator with a confirmation.
 - 2d. System administrator request to 'Delete User':
 - 2d1. System administrator submits search request.
 - 2d2. ITT Tutor System searches for tutor information:
 - 2d1a. Tutor information not found:

ITT Tutor System notifies system administrator [repeat].
 - 2d1b. Tutor information found:

ITT Tutor System presents tutor information:

System administrator confirms it is the desired information and selects to delete.

ITT Tutor System confirms that customer want to delete information.

Cancel deletes operation:

ITT Tutor System aborts the delete operation and notifies system administrator [repeat].

ITT Tutor System deletes the tutoring information and presents the system administrator with a confirmation.
 - 5b. Invalid tutoring information:
 - 5b1. ITT Tutor System notifies system administrator [repeat].

Use Case 2: Locate Tutor

CHARACTERISTIC INFORMATION

Brief description

This use case describes the process of the tutee locating an available tutor

Primary Actor

Tutee

Stakeholders

Tutee – wants to get tutoring.

ITT – wants to improve service to students

Tutor – wants to provide service

Trigger

Tutee accesses ITT Tutor System

Preconditions

Access successful

Guarantees

Success End Condition

Tutee locates an available tutor

Failed End Condition

Tutee unable to locate an available tutor

MAIN SUCCESS SCENARIO

- 1a. Tutee clicks "Find Tutor" link
2. ITT Tutor System returns search options
3. Tutee searches for available tutor
4. ITT Tutor System returns available tutors

5. Tutee writes down contact information
6. Tutee exits system

EXTENSIONS

- *a. Tutee quits:
 - *a1. ITT Tutor System does not make any changes and times out.
- *b. ITT Tutor System fails:
 - *b1. Error message is displayed:
 - *b2. Tutee can try again from step 1.
- 1b. Tutee clicks "More Info." link
 - 1b1. ITT Tutor System displays information topics:
 - 1b2. Tutee reads information and returns.

Use Case 3: Log Session

CHARACTERISTIC INFORMATION

Brief description

This use case describes the process of the tutor logging a tutoring session

Primary Actor

Tutor

Stakeholders

ITT – wants to improve service to students

Tutor – wants to log tutor session

Trigger

Tutor logs in

Preconditions

Log in is successful

Guarantees

Success End Condition

Tutor logs a session

Failed End Condition

Tutor doesn't log a session

MAIN SUCCESS SCENARIO

- 1a. Tutor clicks "Log Session" link
2. ITT Tutor System displays input based on tutor login information
3. Tutor inputs session information
4. ITT Tutor System validates information according to information rules.
- 5a. ITT Tutor System saves session information and presents the tutor with a confirmation.

EXTENSIONS

- *a. Tutor quits:
 - *a1. ITT Tutor System does not make any changes and times out:
- *b. ITT Tutor System fails:
 - *b1. Error message is displayed and data is saved if possible:
 - *b2. Tutor can try again from step 1.
- 1b. Tutor clicks "More Info." Link:
 - 1b1. ITT Tutor System displays information topics:
 - 1b2. Tutor reads information and returns.
- 5b. Invalid tutoring information:
 - 5b1. ITT Tutor System notifies tutor [repeat].

Use Case 4: Rate Tutor

CHARACTERISTIC INFORMATION

Brief description

This use case describes the process of the tutee rating a tutor

Primary Actor

Tutee

Stakeholders

ITT – wants to improve service to students

Tutee – wants to rate tutor

Trigger

Tutee logs in

Preconditions

Log in is successful

Guarantees

Success End Condition

Tutee rates a tutor

Failed End Condition

Tutee doesn't rate a tutor

MAIN SUCCESS SCENARIO

- 1a. Tutee clicks "Rate Tutor" link
2. ITT Tutor System displays input based on tutee login information
3. Tutee inputs session information
4. ITT Tutor System validates information according to information rules.
- 5a. ITT Tutor System saves session information and presents the tutee with a confirmation.

EXTENSIONS

- *a. Tutee quits:
 - *a1. ITT Tutor System does not make any changes and times out.
- *b. ITT Tutor System fails:
 - *b1. Error message is displayed and data is saved if possible.
 - *b2. Tutee can try again from step 1.
- 1b. Tutee clicks "More Info." Link:
 - 1b1. ITT Tutor System displays information topics:
 - 1b2. Tutee reads information and returns.
- 5b. Invalid tutoring information:
 - 5b1. ITT Tutor System notifies tutee [repeat].

Use Case 5: Create Profile

CHARACTERISTIC INFORMATION

Brief description

This use case describes the process of creating a profile by the user (tutor or tutee).

Primary Actor

User

Stakeholders

User – wants to utilize the tutoring system

ITT – wants to improve service to students

Trigger

User accesses system

Preconditions

Access successful

Guarantees

Success End Condition

User creates a profile

Failed End Condition

User unable to create a profile

MAIN SUCCESS SCENARIO

- 1a. User clicks "Create Profile" link
2. ITT Tutor System displays input based on user login information
3. User inputs session information
4. ITT Tutor System validates information according to information rules.
- 5a. ITT Tutor System saves session information and presents the user with a confirmation.

EXTENSIONS

- *a. User quits:
 - *a1. ITT Tutor System does not make any changes and times out.
- *b. ITT Tutor System fails:
 - *b1. Error message is displayed:
 - * b2. User can try again from step 1.
- 1b. User clicks "More Info." Link:
 - 1b1. ITT Tutor System displays information topics:
 - 1b2. User reads information and returns.

Use Case 6: Edit Profile**CHARACTERISTIC INFORMATION****Brief description**

This use case describes the process of editing a profile by the user (tutor or tutee).

Primary Actor

User

Stakeholders

User – wants to utilize the tutoring system

ITT – wants to improve service to students

Trigger

User accesses system

Preconditions

Access successful

Guarantees

Success End Condition

User edits a profile

Failed End Condition

User unable to edit a profile

MAIN SUCCESS SCENARIO

- 1a. User clicks "Edit Profile" link
2. ITT Tutor System displays input based on user login information
3. User inputs session information
4. ITT Tutor System validates information according to information rules.
- 5a. ITT Tutor System saves session information and presents the user with a confirmation.

EXTENSIONS

- *a. User quits:
 - *a1. ITT Tutor System does not make any changes and times out.
- *b. ITT Tutor System fails:
 - *b1. Error message is displayed:
 - * b2. User can try again from step 1.
- 1b. User clicks "More Info." Link:
 - 1b1. ITT Tutor System displays information topics:
 - 1b2. User reads information and returns.

Use Case 7: Log user

CHARACTERISTIC INFORMATION

Brief description

This use case verifies that the user (tutee, tutor, or system administrator) has a valid username and password to access the system.

Primary Actor

User

Stakeholders

User

Trigger

A user accesses the ITT Tutoring System and wishes to log on.

Preconditions

None.

Guarantees

Success End Condition

User is allowed to access restricted parts of the system.

Failed End Condition

System logs failed logon attempts.

MAIN SUCCESS SCENARIO

- 1a. User enters username and password.
2. ITT Tutor System validates username and password.
- 3a. ITT Tutor System logs a successful entry, reads user's permissions, and presents the user with a welcome message.

EXTENSIONS

- *a. User decides to quit:
 - *a1. ITT Tutor System restores initial state [fail].
- 1b. User has not created profile:
 - 1b1. User clicks on "Create Profile" and fills out required information:
 - 1b2. ITT Tutor System verifies and stores information and returns to login page (return).
- 3b. User username is invalid:
 - 3b1. ITT Tutor System notifies user and offers user to register as a new user or retry.
 - User request to reenter username:
 - ITT Tutor System requests username and password. [repeat at 1].
 - New user:
 - User performs register new user [repeat at 1].
- 3c. User username is correct, but password is invalid:
 - 3c1. ITT Tutor System requests username and password [repeat at 1].
- 3d. User username is correct, but password is invalid (3 times):
 - 3d1. ITT Tutor System disables access to user (temporarily) [fail].

Use Case Rules

Maintain Information Rules

NUMBER	Rule Description: The System shall
MI1	Set access based on login
MI2	

Locate Tutor Rules

NUMBER	Rule Description: The System shall
LT1	
LT2	

Log Session Rules

NUMBER	Rule Description: The System shall
LS1	Set access based on login
LS2	

Rate Tutor Rules

NUMBER	Rule Description: The System shall
RT1	Set access based on login
RT2	

Create Profile Rules

NUMBER	Rule Description: The System shall
CP1	
CP2	

Edit Profile Rules

NUMBER	Rule Description: The System shall
EP1	Set access based on login
EP2	

Log User Rules

NUMBER	Rule Description: The System shall
LU1	System verifies user is part of ITT Technical Institute
LU2	

APPENDIX D: SQL STATEMENTS

This appendix includes the various SQL statements used to query the *tutorsystem* database.

create_tutor_profile.php

```
include("../includes/db.php");

MYSQL_CONNECT(HOST,USER,PASS) OR DIE("Unable to connect to database");

@mysql_select_db(DB) or die("Unable to select database");

$query=("select * from course");

$result=mysql_query($query) or die ("Unable to Make the Query:" . mysql_error() );

while($row=mysql_fetch_array($result));
```

tutee_profile.php

```
$query = "INSERT INTO student_information
(first_name,last_name,home_phone,email,gender,major,student_id,disposition) VALUES
('$firstname','$lastname','$phone','$email','$gender','$major','$studentid','$disposition')";

;

$result = mysql_query($query, $link);

$query2 ="INSERT INTO login (student_id,username,password) VALUES
('$studentid','$username','$password')";

$result = mysql_query($query2, $link);
```

tutor_profile.php

```

$query = "INSERT INTO student_information
(first_name,last_name,home_phone,email,gender,major,disposition,availability,student_id)
VALUES ('$firstname','$lastname','$phone','$email','$gender','$major',
'$disposition','$availability','$studentid')";
$result = mysql_query($query, $link);

$query2 ="INSERT INTO login (student_id,username,password) VALUES
('$studentid','$username','$password')";
$result = mysql_query($query2, $link);

$sql = "INSERT into teachable_courses (Student_ID,Course_Name) VALUES ('$studentid',
'$coursename')";
$result = mysql_query($sql);

```

edit_tutee_profile.php

```

$query = "UPDATE student_information
SET student_id='$studentid',First_Name='$firstname',Last_Name='$lastname',Home_Phone='$phone',email='$email',Gender='$gender',Major='$major'
WHERE Student_ID = '$studentid'";
$result = mysql_query($query, $link);

$query2 ="UPDATE login SET
student_ID='$studentid',username='$username',password='$password' WHERE
student_id='$studentid'";
$result = mysql_query($query2, $link);

```

login2.php

```
$sql = "SELECT * FROM login WHERE Username = '$username' AND Password =  
'$password' AND Student_ID = '$studentid'";  
  
$result = mysql_query( $sql );
```

populate_tutee_profile.php

```
$sql = mysql_query ("SELECT  
student_information.Student_ID,student_information.Last_Name,student_information.First_  
Name,student_information.Gender,student_information.Home_Phone,student_information.E  
mail,student_information.Major,  
login.Username,login.Password,login.student_id FROM student_information,login WHERE  
student_information.student_id LIKE '$studentid' AND login.student_id LIKE  
'$studentid'");  
  
$row = mysql_fetch_array($sql);
```

edit_tutor_profile.php

```
$query = "UPDATE student_information  
SETstudent_id='$studentid',First_Name='$firstname',Last_Name='$lastname',  
Home_Phone='$phone',email='$email',Gender='$gender',Major='$major',  
availability='$availability'  
WHERE Student_ID = '$studentid'";  
  
$result = mysql_query($query, $link);
```

```
$query2 ="UPDATE login SET
student_ID='$studentid',username='$username',password='$password' WHERE
student_id='$studentid'";
$result = mysql_query($query2, $link);
$sql = "UPDATE teachable_courses SET
student_ID='$studentid',Course_Name='$coursename' WHERE Student_ID='$studentid'";
$result = mysql_query($sql);
```

locate_tutor_course.php

```
$query=("select * from course");
$result=mysql_query($query);
```

query_tutor.php

```
$result = mysql_query( "SELECT
student_information.Last_Name,student_information.First_Name,student_information.Email
,student_information.Home_Phone,student_information.Cell_Phone,student_information.Ma
jor,student_information.Gender
FROM student_information WHERE Last_Name LIKE '$lastname%' AND disposition =
'tutor'");
```

query_tutor_course.php

```
$result = mysql_query( "SELECT
student_information.Last_Name,student_information.First_Name,student_information.Email
```

```
,student_information.Home_Phone,student_information.Cell_Phone,student_information.Ma  
jor,student_information.Gender, teachable_courses.Course_Name  
FROM student_information,teachable_courses WHERE  
student_information.disposition='tutor' AND teachable_courses.Course_Name  
='$coursename' AND student_information.student_id = teachable_courses.student_id");
```

rate_tutor.php

```
$query = "INSERT INTO rate_tutor (tutor,tutee,course,comments) VALUES  
('$tutor','$tutee','$course','$comments')";  
$result = mysql_query($query, $link);
```

rate_tutor_form.php

```
$query=("select Last_Name from student_information WHERE disposition LIKE 'tutor'");  
$result=mysql_query($query);
```

log_tutor_session.php

```
$query = "INSERT INTO tutor_session (hour, minute ,day, month,  
year,tutee,tutor,comments,course) VALUES  
('$hour','$minute','$day','$month','$year','$tutee','$tutor','$comments','$course_name')";  
$result = mysql_query($query, $link);
```

session.php


```
$query=("select Last_Name from student_information WHERE disposition LIKE 'tutee'");  
$result=mysql_query($query);
```

find_user.php

```
$studentid = $_HTTP_POST_VARS['studentid'];  
$db="wyatt1_tutorsystem";  
$link = mysql_connect("localhost", "wyatt1_itt", "student");  
if (! $link)  
die("Couldn't connect to MySQL");  
mysql_select_db($db , $link)  
or die("Couldn't open $db: ".mysql_error());  
  
$sql = mysql_query ("SELECT  
student_information.Student_ID,student_information.Last_Name,student_information.First_  
Name,student_information.Gender,student_information.Home_Phone,student_information.E  
mail,student_information.Major,student_information.availability,  
login.Username,login.Password,login.student_id FROM student_information,login WHERE  
student_information.student_id LIKE '$studentid'");
```

edit_user_profile

```
$query = "UPDATE student_information
```

SET

```
student_id='$studentid',First_Name='$firstname',Last_Name='$lastname',Home_Phone='$  
phone',email='$email',Gender='$gender',Major='$major',availability='$availability'
```

```
WHERE Student_ID = '$studentid'";
```

```
$result = mysql_query($query, $link);
```

```
$query2 ="UPDATE login SET
```

```
student_ID='$studentid',username='$username',password='$password' WHERE  
student_id='$studentid'";
```

```
$result = mysql_query($query2, $link);
```

```
$sql = "UPDATE teachable_courses SET
```

```
student_ID='$studentid',Course_Name='$coursename' WHERE Student_ID='$studentid'";
```

delete_user.php

```
$result = mysql_query( "SELECT student_id, Last_Name, First_Name FROM  
student_information" );
```

delete_record.php

```
mysql_query("DELETE FROM student_information where student_id=$id");
```

retrieve_user.php

```
mysql_query( "SELECT  
student_information.student_id,student_information.Last_Name,student_information.First_N  
ame,student_information.Email,student_information.Home_Phone,student_information.Cell_  
Phone,student_information.Major,student_information.Gender  
FROM student_information");
```

view_sessions.php

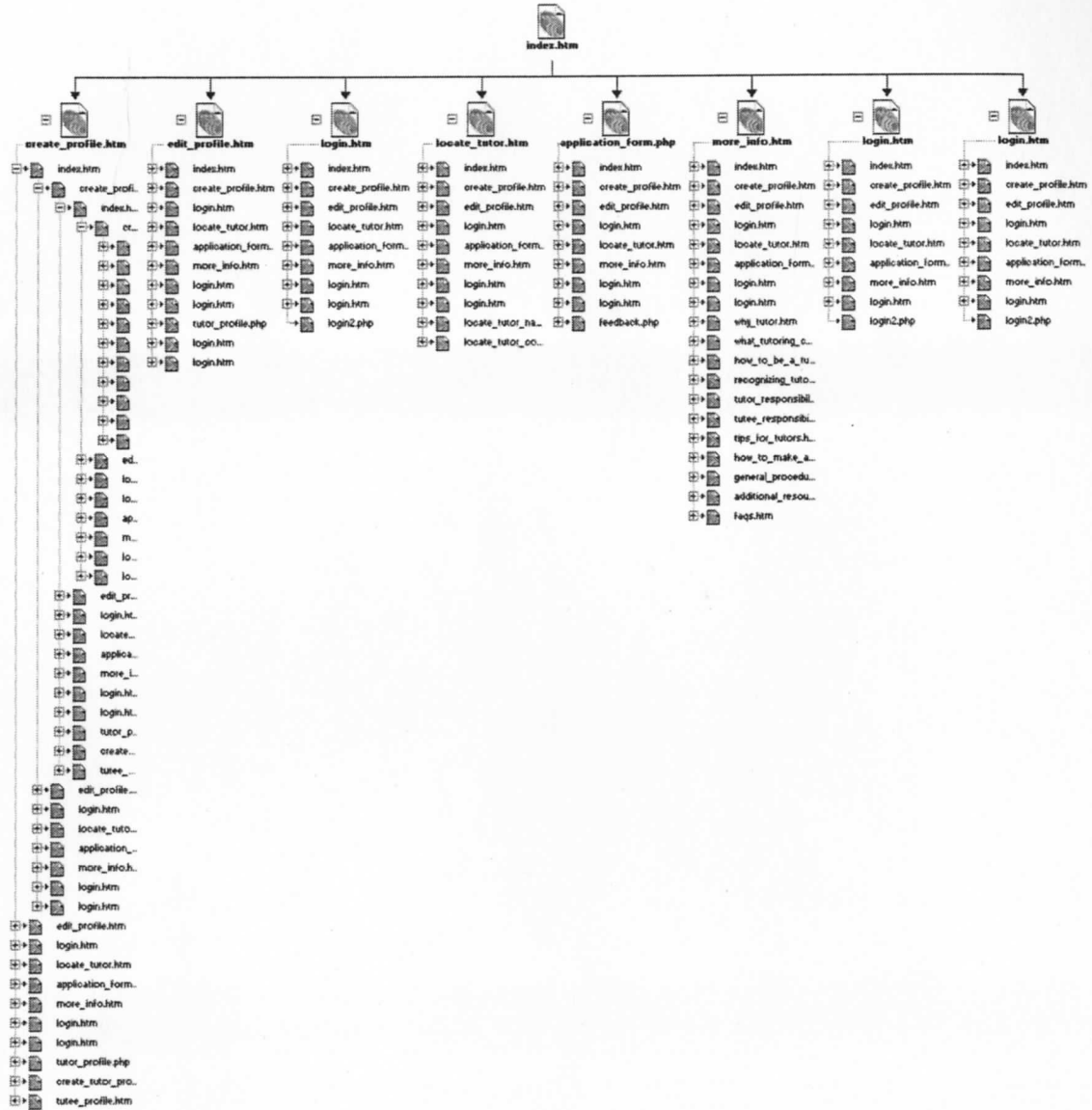
```
$result = mysql_query( "SELECT * FROM tutor_session");
```

view_ratings.php

```
$result = mysql_query( "SELECT * FROM rate_tutor");
```

APPENDIX E: SITE MAP

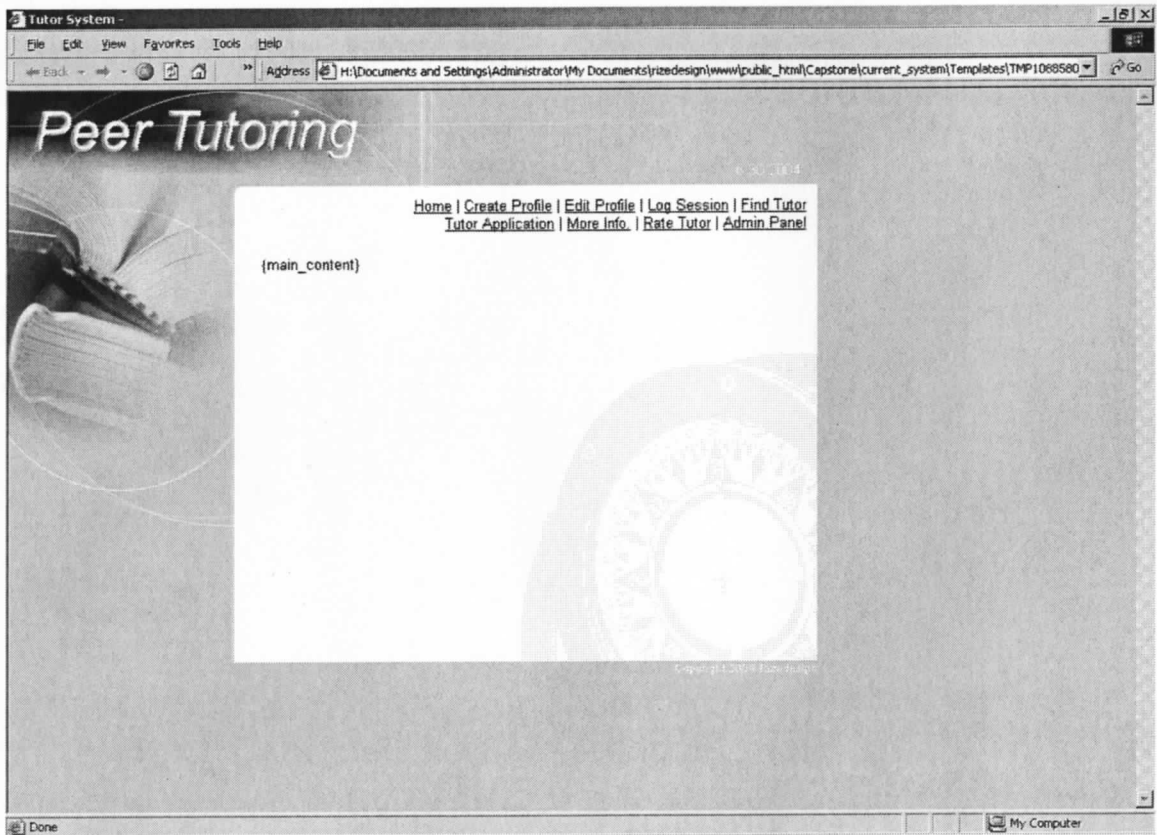
This appendix shows the site map for the ITT Tutor System.



APPENDIX F: SCREEN SHOTS AND SOURCE CODE

This appendix includes all the ITT Tutor System source code and related screen shots. The index.dwt file is a Dreamweaver template used as the basis of design for the entire system. The entirety of HTML is displayed for this template file. Each additional page reflects the change in code of the main content area denoted between the comments `<!-- #BeginEditable "doctitle" -->` and `<!-- #EndEditable -->`.

index.dwt (Dreamweaver template file)



```
<html>
<head>
<!-- #BeginEditable "doctitle" -->
<title>Tutor System</title>
<!-- #EndEditable -->
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link rel="stylesheet" href="../css.css" type="text/css">
```

```

</head>
<body leftmargin="0" topmargin="0" marginwidth="0" marginheight="0"
background="../images/backcolor.jpg">
<table width="720" border="0" cellspacing="0" cellpadding="0">
<tr align="left" valign="top">
<td>
<table cellpadding="0" cellspacing="0" border="0" width="100%">
<tr>
<td colspan="3">
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr>
<td width="93%"></td>
<td width="7%" align="left" valign="bottom">
<table width="100%" border="0" cellspacing="0" cellpadding="5">
<tr>
<td><font color="#FFFFFF" size="2">
<script type="text/javascript">
var d = new Date()
document.write(d.getMonth() + 1)
document.write(".")
document.write(d.getDate())
document.write(".")
document.write(d.getFullYear())
</script>
</font><font size="2" face="Arial, Helvetica, sans-serif" color="#FFFFFF"></font></td>
<td><font size="2" face="Arial, Helvetica, sans-serif" color="#FFFFFF"></font></td>
</tr>
</table>
</td>
</tr>
</table>
<font color="#FFFFFF" size="2"> </font></td>
</tr>
</table>
</tr>
<tr align="left" valign="top">
<td>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr>
<td width="201" align="left" valign="top"></td>
<td align="left" valign="top">
<table width="100%" border="0" cellspacing="0" cellpadding="0" bgcolor="#FFFFFF"
background="../images/compass.jpg">
<tr>

```

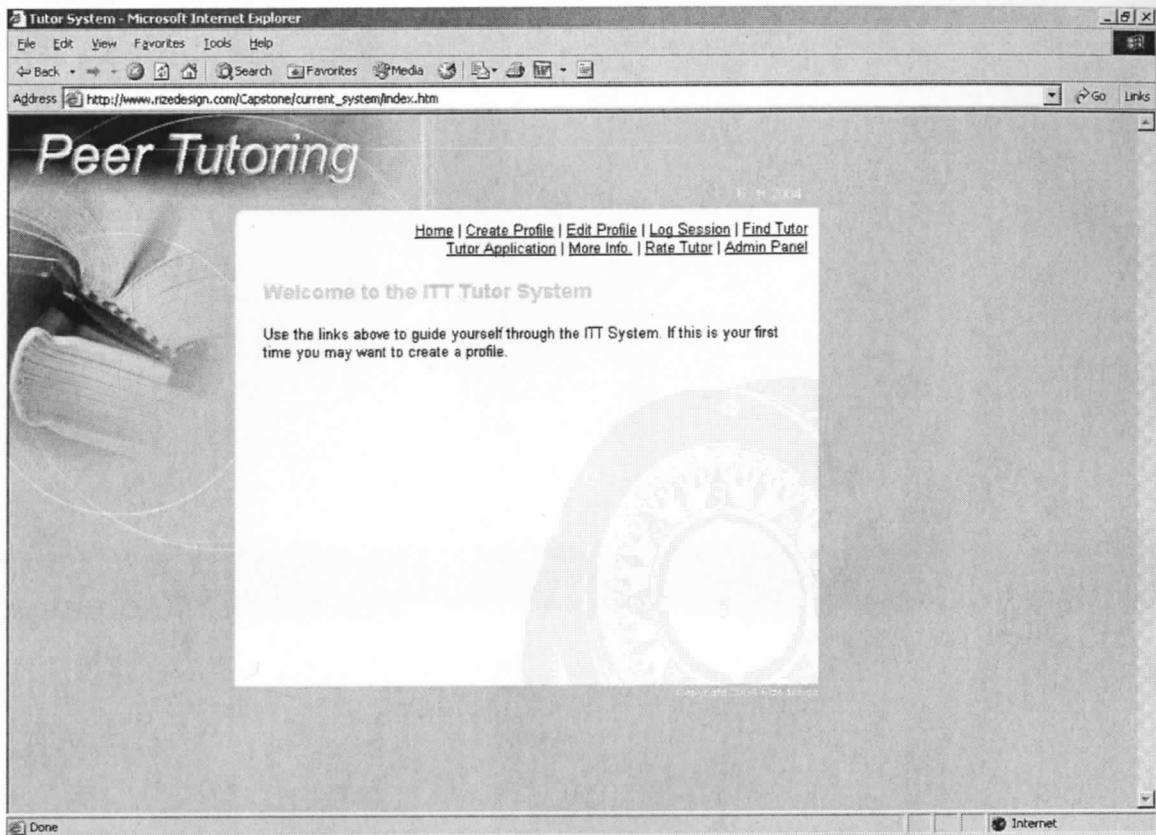
```

<td align="left" valign="top" width="14"><br>

</td>
<td align="left" valign="top">
<table width="100%" border="0" cellspacing="0" cellpadding="10">
<tr>
<td height="39" align="left" valign="top">
<div align="right"><font size="2"> <a href="../index.htm">Home</a>
| <a href="../create_profile/create_profile.htm">Create
Profile</a> | <a href="../edit_profile/edit_profile.htm">Edit
Profile</a> | <a href="../session/login.htm">Log Session</a>
| <a href="../locate_tutor/locate_tutor.htm">Find Tutor</a>
<br>
<a href="../application/application_form.php">Tutor
Application</a> | <a href="../more_info/more_info.htm">More
Info.</a> | <a href="../ratings/login.htm">Rate Tutor</a>
| <a href="../admin/login.htm">Admin Panel</a></font></div>
</td>
</tr>
<tr align="left" valign="top">
<td><!-- #BeginEditable "main_content" -->{main_content}<!-- #EndEditable -->
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
<tr align="right">
<td><font color="#FFFFFF" size="1">Copyright 2004 Rizedesign</font></td>
</tr>
</table>
</body>
</html>

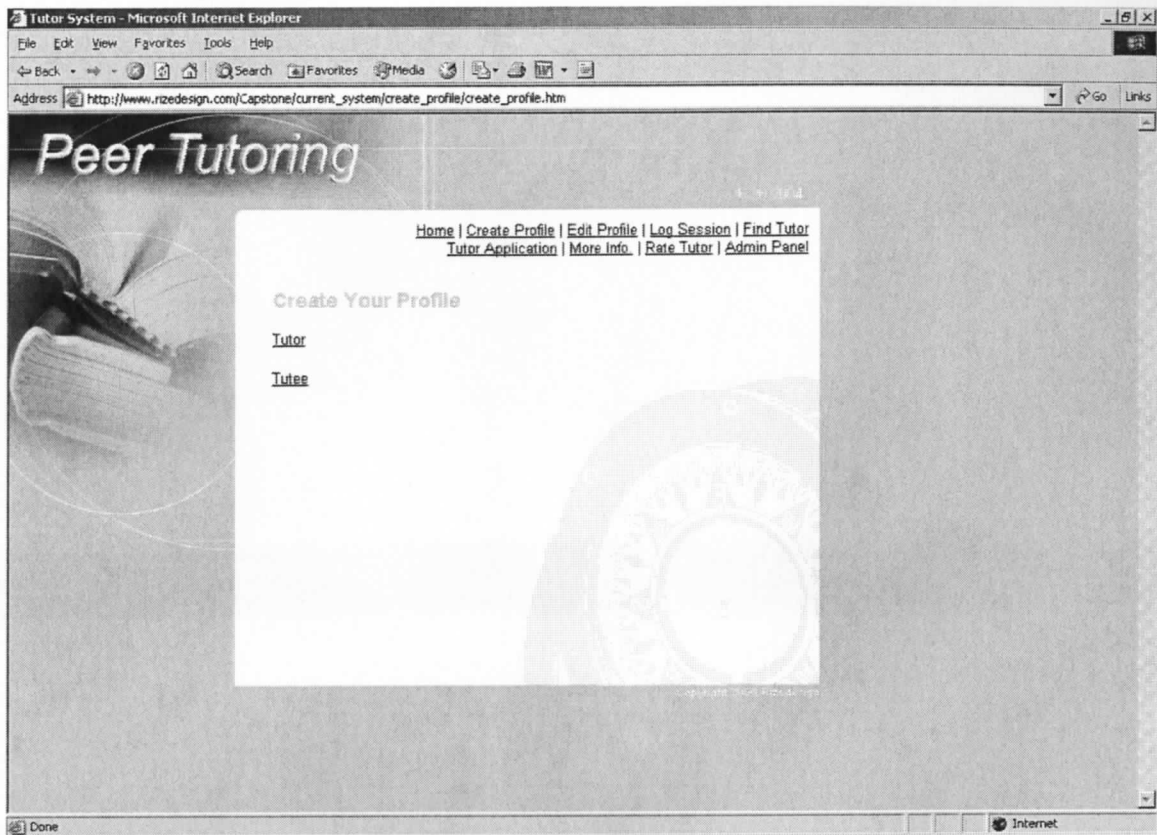
```

index.htm



```
<!-- #BeginEditable "main_content"-->  
<p><font color="#CCCCCC"><b><font size="4">Welcome to the ITT Tutor  
System</font></b></font></p>  
<p>Use the links above to guide yourself through the ITT System. If this is your first time  
you may want to create a profile.</p>  
<p>&nbsp;</p>  
<p>&nbsp;</p>  
<p align="right">&nbsp;</p>  
<!-- #EndEditable -->
```


create_profile.htm



```
<!-- #BeginEditable "main_content" -->
<form name="form1" method="post" action="tutor_profile.php">
<table width="348" border="0" cellspacing="0" cellpadding="8">
<tr>
<td colspan="2">
<p><b><font color="#CCCCCC" size="4">Create Your Profile</font></b></p>
</td>
</tr>
<tr>
<td colspan="2">
<p><a href="create_tutor_profile.php">Tutor</a></p>
<p><a href="tutee_profile.htm">Tutee</a></p>
</td>
</tr>
</table>
</form>
<!-- #EndEditable -->
```

create_tutor_profile.php

Home | Create Profile | Edit Profile | Log Session | Find Tutor
Tutor Application | More Info | Rate Tutor | Admin Panel

Tutor Profile

Student ID:*

Last Name:*

First Name:*

Gender:*

Phone:*

Email:*

Major:*

Availability:*

Teachable Courses:*
 College Mathematics I
 Problem Solving
 College Mathematics II
 Composition II

Username:*

Password:*

Fields marked with an asterisk (*) are required.

```
<form name="form1" method="post" action="tutor_profile.php" onSubmit="return
validate_form ( );">
<table width="349" border="0" cellspacing="0" cellpadding="5">
<tr>
<td colspan="2">
<p><b><font color="#CCCCCC" size="4">Tutor Profile</font></b></p>
</td>
</tr>
<tr>
<td>Student ID:*</td>
<td>
<input type="text" name="studentid">
</td>
</tr>
<tr>
<td>Last Name:*</td>
<td>
<input type="text" name="lastname">
</td>
</tr>
<tr>
<td>First Name:*</td>
```

```

<td>
<input type="text" name="firstname">
</td>
</tr>
<tr>
<td>Gender:* </td>
<td>
<select name="gender">
<option selected>Male</option>
<option>Female</option>
</select>
</td>
</tr>
<tr>
<td>Phone:* </td>
<td>
<input type="text" name="phone" >
</td>
</tr>
<tr>
<td>Email:* </td>
<td>
<input type="text" name="email">
</td>
</tr>
<tr>
<td>Major:* </td>
<td>
<select name="major">
<option selected>MM</option>
<option>CEET</option>
<option>SAP</option>
<option>CADD</option>
<option>WEB</option>
</select>
</td>
</tr>
<tr>
<td align="left">Availability:* </td>
<td align="left" valign="top">
<input type="text" name="availability" value="">
</td>
</tr>
<tr>
<td align="left">Teachable Courses:* </td>
<td>

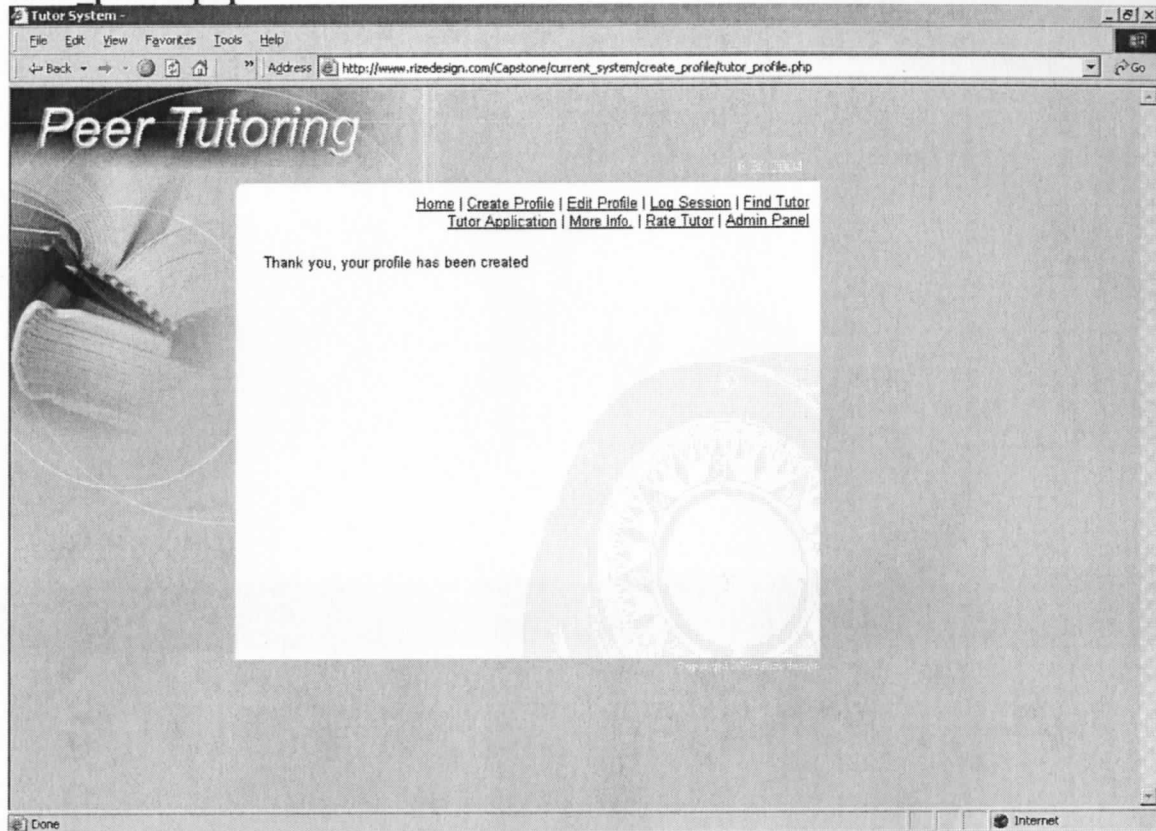
```

```

<select name="coursename[]" multiple size=5>
<?php
include("../includes/db.php");
MYSQL_CONNECT(HOST,USER,PASS) OR DIE("Unable to connect to database");
@mysql_select_db(DB) or die("Unable to select database");
$query=("select * from course");
$result=mysql_query($query) or die("Unable to Make the Query:" . mysql_error() );
while($row=mysql_fetch_array($result)){
echo "<OPTION VALUE=".$row['Course_ID'].">".$row['Course_Name'].</OPTION>";
}
?>
</select>
</td>
</tr>
<tr>
<td>Username: *</td>
<td>
<input type="text" name="username" >
</td>
</tr>
<tr>
<td>Password: *</td>
<td>
<input type="password" name="password" >
</td>
</tr>
<tr>
<td>&nbsp;</td>
<td>
<input type="hidden" name="disposition" value="tutor">
<input type="submit" name="Submit" value="Submit">
<input type="reset" name="Submit2" value="Reset">
</td>
</tr>
<tr>
<td colspan="2"><font size="1">Fields marked with
an asterisk (*) are required.</font></td>
</tr>
</table>
</form>

```

tutor_profile.php



```
<?php session_start(); ?>
<!-- #BeginEditable "main_content" -->
<?php
$studentid = $HTTP_POST_VARS['studentid'];
$firstname = $HTTP_POST_VARS['firstname'];
$lastname = $HTTP_POST_VARS['lastname'];
$phone = $HTTP_POST_VARS['phone'];
$email = $HTTP_POST_VARS['email'];
$gender = $HTTP_POST_VARS['gender'];
$major = $HTTP_POST_VARS['major'];
$disposition = $HTTP_POST_VARS['disposition'];
$availability = $HTTP_POST_VARS['availability'];
$courses = serialize($_POST['courses']);
$username = $HTTP_POST_VARS['username'];
$password = $HTTP_POST_VARS['password'];
$coursename = implode(' ', $_POST['coursename']);

$db="wyatt1_tutorsystem";
$link = mysql_connect("localhost", "wyatt1_itt", "student");
if (! $link)
die("Couldn't connect to MySQL");
mysql_select_db($db, $link)
or die("Couldn't open $db: ".mysql_error());
```

```

if (!$studentid && !$firstname && !$lastname && !$phone && !$email && !$gender &&
!$major && !$disposition && !$availability)
{
    echo "you didn't fill in all the required fields! go back and do it again...";
}

else
{
$query = "INSERT INTO student_information
(first_name,last_name,home_phone,email,gender,major,disposition,availability,student_id)
VALUES
('$firstname','$lastname','$phone','$email','$gender','$major','$disposition','$availability'
,$studentid)";
$result = mysql_query($query, $link);
$query2 ="INSERT INTO login (student_id,username,password) VALUES
('$studentid','$username','$password')";
$result = mysql_query($query2, $link);
$sql = "INSERT into teachable_courses (Student_ID,Course_Name) VALUES ('$studentid',
'$coursename')";
$result = mysql_query($sql);
echo "Thank you, your profile has been created";
}
session_destroy();
?>
<!-- #EndEditable -->

```

tutee_profile.htm

Peer Tutoring

Home | [Create Profile](#) | [Edit Profile](#) | [Log Session](#) | [Find Tutor](#)
[Tutor Application](#) | [More Info](#) | [Rate Tutor](#) | [Admin Panel](#)

Tutee Profile

Student ID:*

Last Name:*

First Name:*

Gender:*

Phone:*

Email:*

Major:*

Username:*

Password:*

Fields marked with an asterisk (*) are required.

```
<?php session_start(); ?>
<!-- #BeginEditable "main_content" -->
<form name="form1" method="post" action="tutee_profile.php" onSubmit="return
validate_form ();">
<table width="349" border="0" cellspacing="0" cellpadding="8">
<tr>
<td colspan="2">
<p><b><font color="#CCCCCC" size="4">Tutee Profile</font></b></p>
</td>
</tr>
<tr>
<td>Student ID:*</td>
<td>
<input type="text" name="studentid">
</td>
</tr>
<tr>
<td>Last Name:*</td>
<td>
<input type="text" name="lastname">
</td>
</tr>
```

```
<tr>
<td>First Name:*/td>
<td>
<input type=""text"" name=""firstname"">
</td>
</tr>
<tr>
<td>Gender:*/td>
<td>
<select name=""gender"">
<option selected>Male</option>
<option>Female</option>
</select>
</td>
</tr>
<tr>
<td>Phone:*/td>
<td>
<input type=""text"" name=""phone"" >
</td>
</tr>
<tr>
<td>Email:*/td>
<td>
<input type=""text"" name=""email"">
</td>
</tr>
<tr>
<td>Major:*/td>
<td>
<select name=""major"">
<option selected>MM</option>
<option>CEET</option>
<option>SAP</option>
<option>CADD</option>
<option>WEB</option>
</select>
</td>
</tr>
<tr>
<td>Username:*/td>
<td>
<input type=""text"" name=""username"" >
</td>
</tr>
<tr>
```

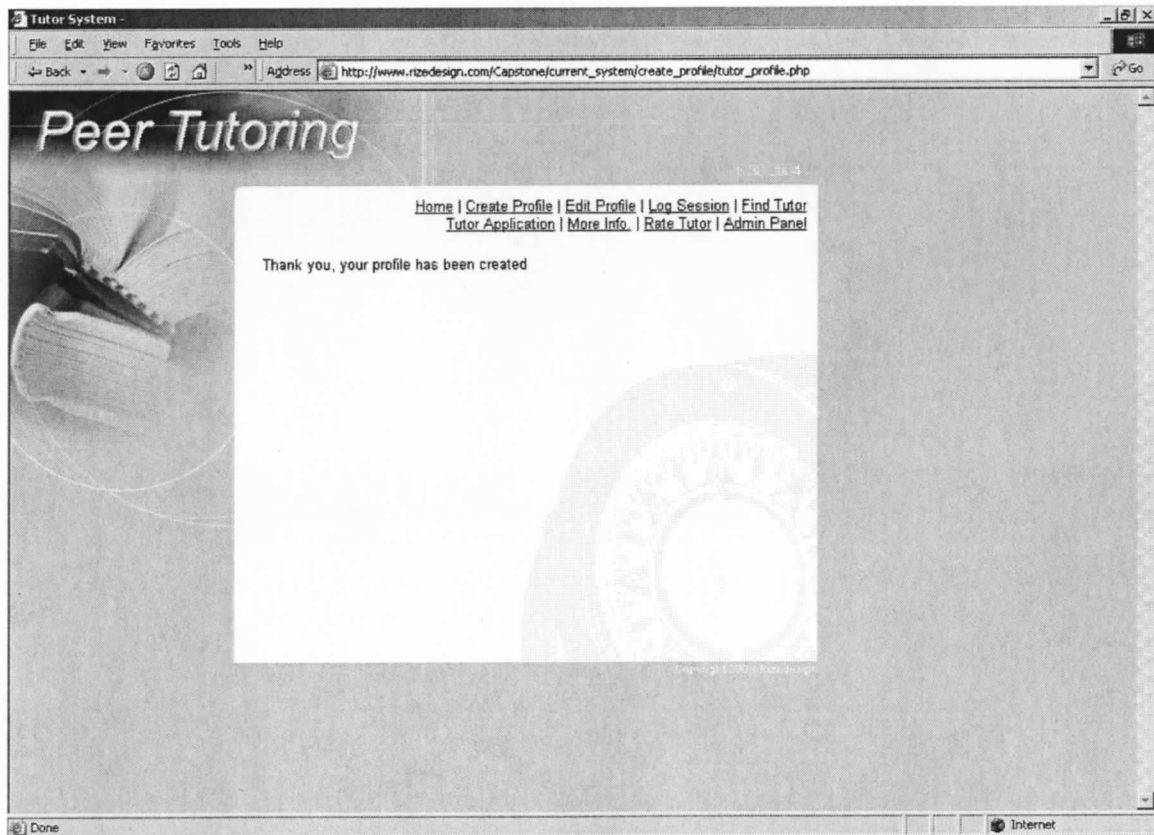


```

<td>Password:*</td>
<td>
<input type="password" name="password" >
</td>
</tr>
<tr>
<td>&nbsp;</td>
<td> <input type="hidden" name="disposition" value="tutee">
<input type="submit" name="Submit" value="Submit">
<input type="reset" name="Submit2" value="Reset">
</td>
</tr>
<tr>
<td colspan="2"><font size="2">Fields marked with an asterisk (*) are
required.</font></td>
</tr>
</table>
</form>
<!-- #EndEditable -->

```

tutee_profile.php



```

<!-- #BeginEditable "main_content" -->
<?php

```

```

$studentid = $HTTP_POST_VARS['studentid'];
$firstname = $HTTP_POST_VARS['firstname'];
$lastname = $HTTP_POST_VARS['lastname'];
$phone = $HTTP_POST_VARS['phone'];
$email = $HTTP_POST_VARS['email'];
$gender = $HTTP_POST_VARS['gender'];
$major = $HTTP_POST_VARS['major'];
$username = $HTTP_POST_VARS['username'];
$password = $HTTP_POST_VARS['password'];
$disposition = $HTTP_POST_VARS['disposition'];

$db="wyatt1_tutorsystem";
$link = mysql_connect("localhost", "wyatt1_itt", "student");
if (! $link)
die("Couldn't connect to MySQL");
mysql_select_db($db, $link)
or die("Couldn't open $db: ".mysql_error());

if (!$studentid && !$firstname && !$lastname && !$phone && !$email && !$gender &&
!$major && !$disposition && !$availability)
{
echo "you didn't fill in all the required fields! go back and do it again...";
}

else
{
$query = "INSERT INTO student_information
(first_name,last_name,home_phone,email,gender,major,student_id,disposition) VALUES
('$firstname','$lastname','$phone','$email','$gender','$major','$studentid','$disposition')";
;
$result = mysql_query($query, $link);
$query2 ="INSERT INTO login (student_id,username,password) VALUES
('$studentid','$username','$password')";
$result = mysql_query($query2, $link);
echo "Thank you, your profile has been created";
}
?>
<!-- #EndEditable -->

```

edit_tutee_profile.php

Peer Tutoring

Home | [Create Profile](#) | [Edit Profile](#) | [Log Session](#) | [Find Tutor](#)
[Tutor Application](#) | [More Info](#) | [Rate Tutor](#) | [Admin Panel](#)

Edit Your Profile

Student ID:

Last Name:

First Name:

Gender:

Major:

Phone:

Email:

Username:

Password:

Fields marked with an asterisk(*) are required.

```
<?php session_start();?>
```

```
<!-- #BeginEditable "main_content" -->
```

```
<?php
```

```
$studentid = $HTTP_POST_VARS['studentid'];  
$firstname = $HTTP_POST_VARS['firstname'];  
$lastname = $HTTP_POST_VARS['lastname'];  
$phone = $HTTP_POST_VARS['phone'];  
$email = $HTTP_POST_VARS['email'];  
$gender = $HTTP_POST_VARS['gender'];  
$major = $HTTP_POST_VARS['major'];  
$username = $HTTP_POST_VARS['username'];  
$password = $HTTP_POST_VARS['password'];
```

```
$db="wyatt1_tutorsystem";
```

```
$link = mysql_connect("localhost", "wyatt1_itt", "student");
```

```
if (! $link)
```

```
die("Couldn't connect to MySQL");
```

```
mysql_select_db($db, $link)
```

```
or die("Couldn't open $db: ".mysql_error());
```

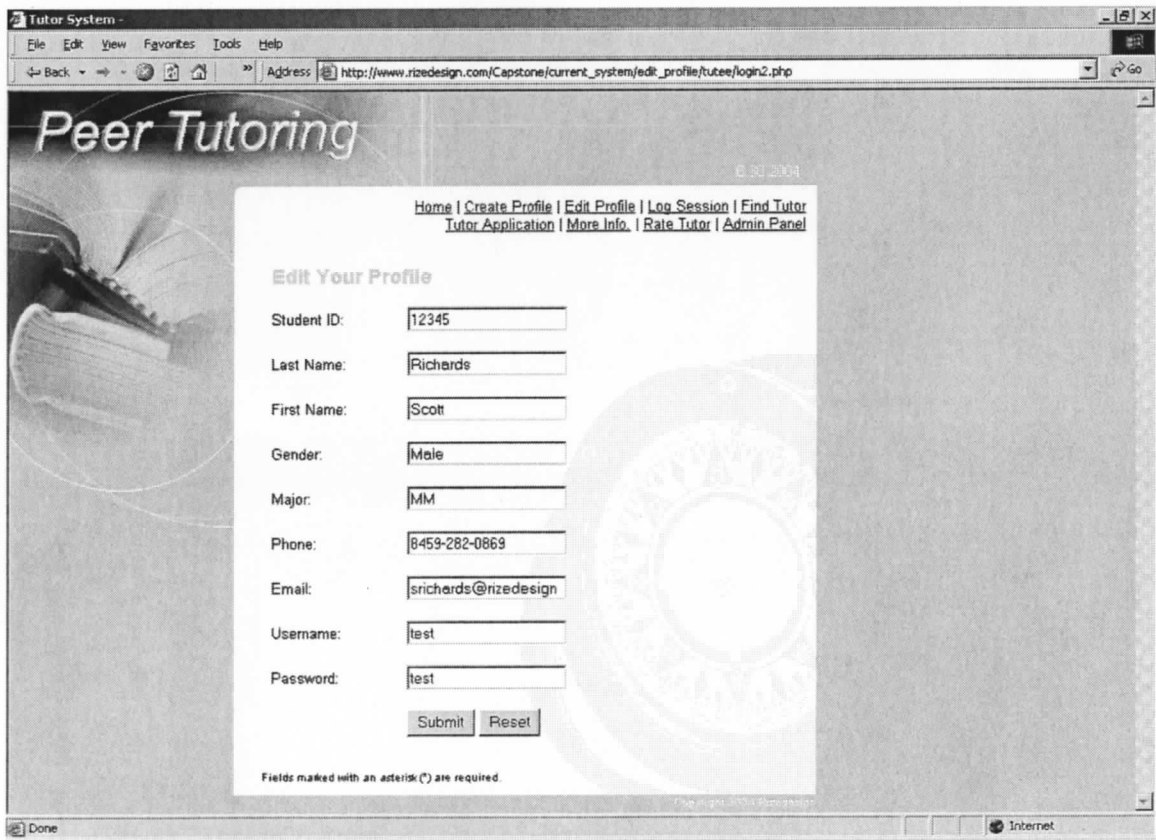
```
if (!$studentid && !$firstname && !$lastname && !$phone && !$email && !$gender &&  
!$major && !$disposition && !$availability)
```

```

{
    echo "you didn't fill in all the required fields! go back and do it again...";
}
else
{
    $query = "UPDATE student_information
SET
student_id='$studentid',First_Name='$firstname',Last_Name='$lastname',Home_Phone='$
phone',email='$email',Gender='$gender',Major='$major'
WHERE Student_ID = '$studentid'";
    $result = mysql_query($query, $link);
$query2 ="UPDATE login SET
student_ID='$studentid',username='$username',password='$password' WHERE
student_id='$studentid'";
$result = mysql_query($query2, $link);
echo "Thank you, Your profile has been edited";
}
session_destroy();
?>
<!-- #EndEditable -->

```

populate_tutee_profile.php



```
<?php session_start(); ?>
```

```

<!-- #BeginEditable "main_content" --> <?php
$studentid = $_HTTP_POST_VARS['studentid'];
$db="wyatt1_tutorsystem";
$link = mysql_connect("localhost", "wyatt1_itt", "student");
if (! $link)
die("Couldn't connect to MySQL");
mysql_select_db($db , $link)
or die("Couldn't open $db: ".mysql_error());

$sql = mysql_query ("SELECT
student_information.Student_ID,student_information.Last_Name,student_information.First_
Name,student_information.Gender,student_information.Home_Phone,student_information.E
mail,student_information.Major,
login.Username,login.Password,login.student_id FROM student_information,login WHERE
student_information.student_id LIKE '$studentid' AND login.student_id LIKE
'$studentid'");
$row = mysql_fetch_array($sql);
print "<form action=\"edit_tutee_profile.php\" method=\"post\">";
do
{
print "<p>";
print "<table width=\"349\" border=\"0\" cellspacing=\"0\" cellpadding=\"8\">";
print "<tr>";
print "<td colspan=\"2\">";
print "<b><font color=\"#CCCCCC\" size=\"4\">";
print "Edit Your Profile";
print "</font></b>";
print "</td>";
print "</tr>";
print "<tr>";
print "<td>";
print "Student ID: ";
print "</td>";
print "<td>";
print "<input type=\"text\" name=\"studentid\" value=\"\"";
print $row['Student_ID'];
print ">";
print "</td>";
print "</tr>";
print "<tr>";
print "<td>";
print "Last Name: ";
print "</td>";
print "<td>";
print "<input type=\"text\" name=\"lastname\" value=\"\"";
print $row['Last_Name'];

```

```

print "\>";
print "</td>";
print "</tr>";
print "<tr>";
print "<td>";
print "First Name: ";
print "</td>";
print "<td>";
print "<input type=\"text\" name=\"firstname\" value=\"\"";
print $row['First_Name'];
print "\>";
print "</td>";
print "</tr>";
print "<tr>";
print "<td>";
print "Gender: ";
print "</td>";
print "<td>";
print "<input type=\"text\" name=\"gender\" value=\"\"";
print $row['Gender'];
print "\>";
print "</td>";
print "</tr>";
print "<tr>";
print "<td>";
print "Major: ";
print "</td>";
print "<td>";
print "<input type=\"text\" name=\"major\" value=\"\"";
print $row['Major'];
print "\>";
print "</td>";
print "</tr>";
print "<tr>";
print "<td>";
print "Phone: ";
print "</td>";
print "<td>";
print "<input type=\"text\" name=\"phone\" value=\"\"";
print $row['Home_Phone'];
print "\>";
print "</td>";
print "</tr>";
print "<tr>";
print "<td>";
print "Email: ";

```

```

print "</td>";
print "<td>";
print "<input type='text' name='email' value='\"";
print $row['Email'];
print "\>";
print "</td>";
print "</tr>";
print "<tr>";
print "<td>";
print "Username: ";
print "</td>";
print "<td>";
print "<input type='text' name='username' value='\"";
print $row['Username'];
print "\>";
print "</td>";
print "</tr>";
print "<tr>";
print "<td>";
print "Password: ";
print "</td>";
print "<td>";
print "<input type='text' name='password' value='\"";
print $row['Password'];
print "\>";
print "</td>";
print "</tr>";
print "<tr>";
print "<td>";
print "</td>";
print "<td>";
print "<input type='submit' value='Submit' /> <input type='reset' value='Reset'";
/></table></form>";
} while ($row = mysql_fetch_array($sql));

```

?>

Fields marked with an asterisk (*) are required.

<!-- #EndEditable -->

edit_tutor_profile.php

The screenshot shows a web browser window titled 'Tutor System'. The address bar displays 'http://www.rizedesign.com/Capstone/current_system/edit_profile/tutor/login2.php'. The page content includes a navigation menu with links: Home, Create Profile, Edit Profile, Log Session, Find Tutor, Tutor Application, More Info, Rate Tutor, and Admin Panel. The main heading is 'Edit Your Profile'. The form fields are as follows:

Student ID:	<input type="text" value="12345"/>
Last Name:	<input type="text" value="Richards"/>
First Name:	<input type="text" value="Scott"/>
Gender:	<input type="text" value="Male"/>
Major:	<input type="text" value="MM"/>
Phone:	<input type="text" value="8459-282-0869"/>
Email:	<input type="text" value="srichards@rizedesign"/>
Availability:	<input type="text" value="anytime"/>
Username:	<input type="text" value="test"/>
Password:	<input type="text" value="test"/>

Teachable Courses:

```
<?php session_start();?>
<!-- #BeginEditable "main_content" -->
<?php
$studentid = $HTTP_POST_VARS['studentid'];
$firstname = $HTTP_POST_VARS['firstname'];
$lastname = $HTTP_POST_VARS['lastname'];
$phone = $HTTP_POST_VARS['phone'];
$email = $HTTP_POST_VARS['email'];
$gender = $HTTP_POST_VARS['gender'];
$major = $HTTP_POST_VARS['major'];
$availability = $HTTP_POST_VARS['availability'];
$username = $HTTP_POST_VARS['username'];
$password = $HTTP_POST_VARS['password'];
$coursename = implode(',',$_POST['coursename']);

$db="wyatt1_tutorsystem";
$link = mysql_connect("localhost", "wyatt1_itt", "student");
if (! $link)
die("Couldn't connect to MySQL");
mysql_select_db($db, $link)
or die("Couldn't open $db: ".mysql_error());
```



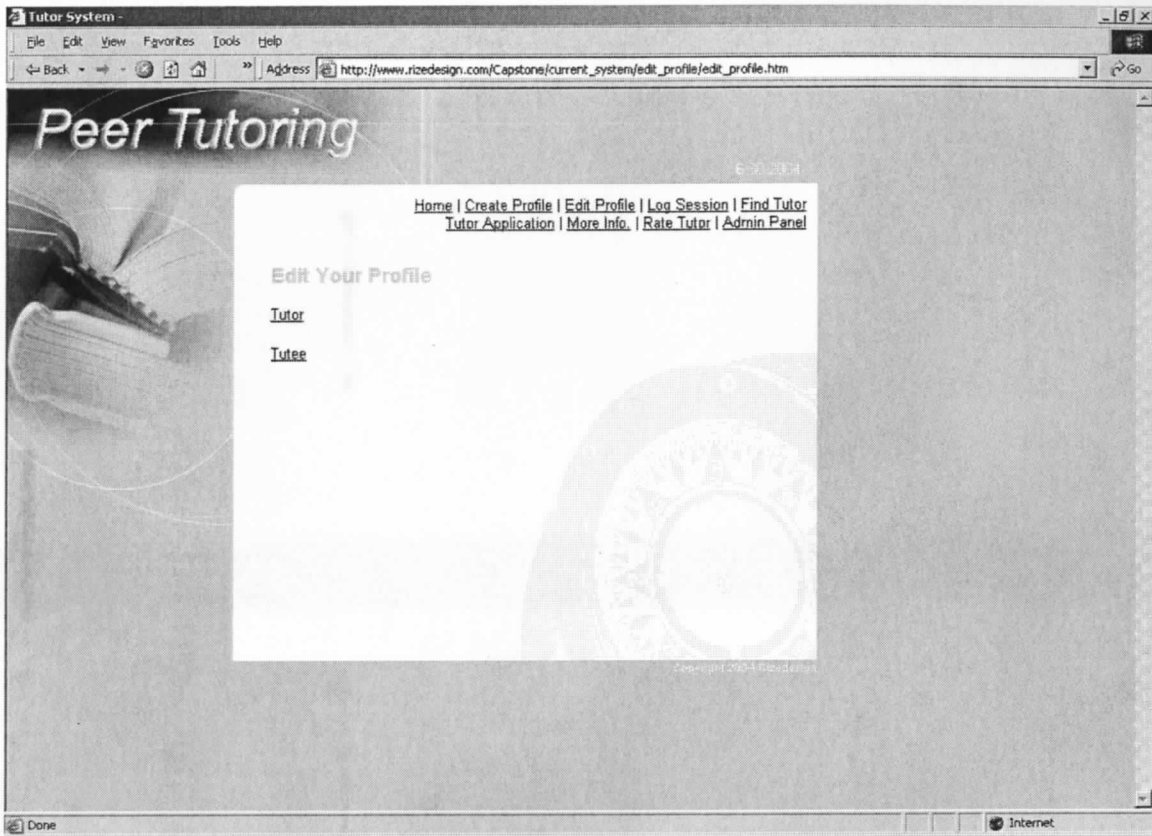
```

// check if the variables exist (if the form fields have been filled in)
if (!$studentid && !$firstname && !$lastname && !$phone && !$email && !$gender &&
!$major && !$disposition && !$availability)
{
    echo "you didn't fill in all the required fields! go back and do it again...";
}
else
{
$query = "UPDATE student_information
SET
student_id='$studentid',First_Name='$firstname',Last_Name='$lastname',Home_Phone='$
phone',email='$email',Gender='$gender',Major='$major',availability='$availability'
WHERE Student_ID = '$studentid'";
$result = mysql_query($query, $link);

$query2 ="UPDATE login SET
student_ID='$studentid',username='$username',password='$password' WHERE
student_id='$studentid'";
$result = mysql_query($query2, $link);
$sql = "UPDATE teachable_courses SET
student_ID='$studentid',Course_Name='$coursename' WHERE Student_ID='$studentid'";
$result = mysql_query($sql);
    echo "Thank you, Your profile has been edited";
}
session_destroy();
?>
<!-- #EndEditable -->

```

edit_profile.htm



```
<!-- #BeginEditable "main_content" -->
<form name="form1" method="post" action="../create_profile/tutor_profile.php">
<table width="348" border="0" cellspacing="0" cellpadding="8">
<tr>
<td colspan="2">
<p><b><font color="#CCCCCC" size="4">Edit Your
Profile</font></b></p>
</td>
</tr>
<tr>
<td colspan="2">
<p><a href="tutor/login.htm">Tutor</a></p>
<p><a href="tutee/login.htm">Tutee</a></p>
</td>
</tr>
</table>
</form>
<!-- #EndEditable -->
```

application_form.php

```

<!-- #BeginEditable "main_content" -->
<table width="100%" border="0" cellspacing="0" cellpadding="8">
<tr>
<td colspan="2">
<p><b><font color="#CCCCCC" size="4">Tutor Application</font></b></p>
</td>
</tr>
<tr align="left" valign="top">
<td colspan="2">
<form method="post" action="feedback.php" onSubmit="return validate_form ();">
<table width="349" border="0" cellspacing="0" cellpadding="8">
<tr>
<td>Student ID:*</td>
<td>
<input type="text" name="txtstudentid">
</td>
</tr>
<tr>
<td>Last Name:*</td>
<td>
<input type="text" name="txtlastname">
</td>

```

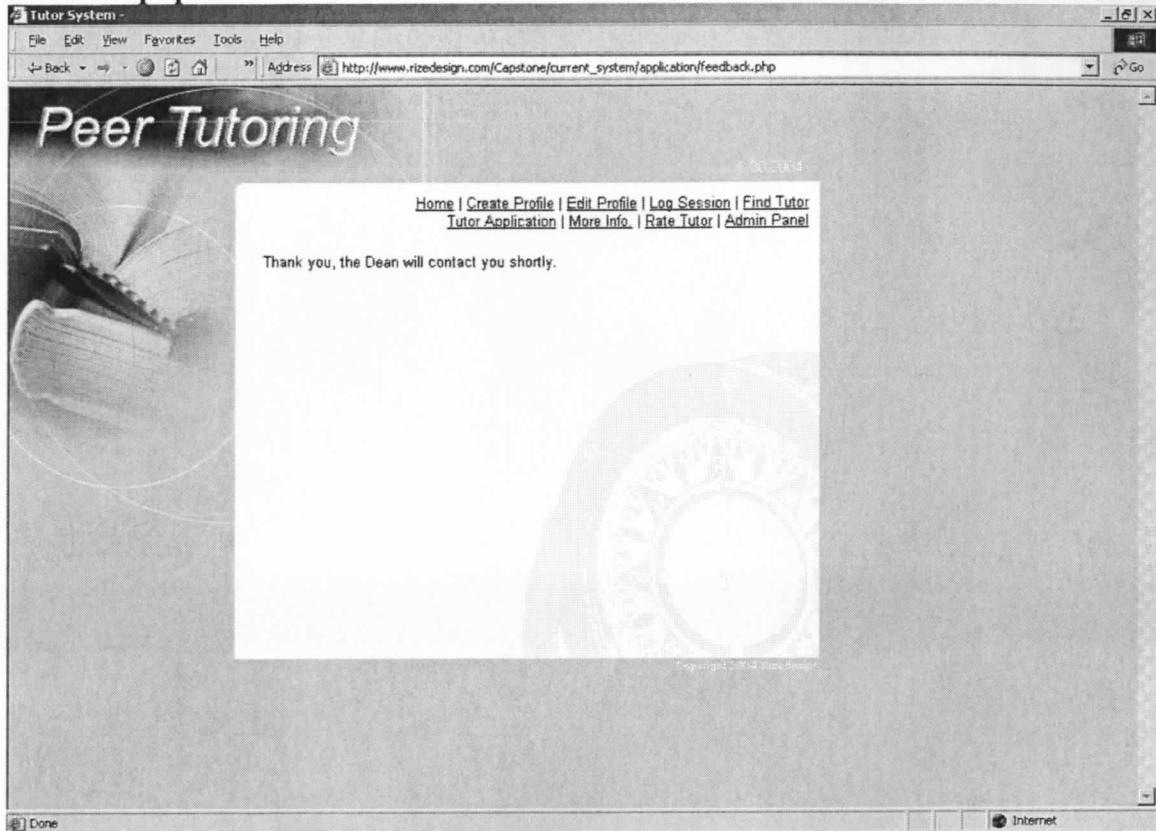
```

</tr>
<tr>
<td>First Name:*</td>
<td>
<input type="text" name="txtfirstname">
</td>
</tr>
<tr>
<td>Gender: * </td>
<td>
<select name="txtgender">
<option selected>Male</option>
<option>Female</option>
</select>
</td>
</tr>
<tr>
<td>Phone:*</td>
<td>
<input type="text" name="txtphone" >
</td>
</tr>
<tr>
<td>Email:*</td>
<td>
<input type="text" name="txtemail">
</td>
</tr>
<tr>
<td>Major:*</td>
<td>
<select name="txtmajor">
<option selected>MM</option>
<option>CEET</option>
<option>SAP</option>
<option>CADD</option>
<option>WEB</option>
</select>
</td>
</tr>
<tr>
<td align="left" valign="top">Year/quarter enrollment:*</td>
<td>
<select name="txtenrollment">
<option selected>Summer 2004</option>
<option>Fall 2004</option>

```

```
<option>Winter 2004</option>
<option>Spring 2005</option>
<option>Summer 2005</option>
<option>Fall 2005</option>
<option>Winter 2005</option>
</select>
</td>
</tr>
<tr>
<td>&nbsp;</td>
<td>
<input type="submit" value="Submit">
<input type="reset" value="Reset">
</td>
</tr>
<tr>
<td colspan="2"><font size="1">Fields marked with an asterisk (*) are
required.</font></td>
</tr>
</table>
<p>&nbsp;</p>
</form>
<p>&nbsp;</p>
</td>
</tr>
</table>
<!-- #EndEditable -->
```

feedback.php



```
<!-- #BeginEditable "main_content" -->
<?php
$studentid = $HTTP_POST_VARS['txtstudentid'];
$lastname = $HTTP_POST_VARS['txtlastname'];
$firstname = $HTTP_POST_VARS['txtfirstname'];
$gender = $HTTP_POST_VARS['txtgender'];
$phone = $HTTP_POST_VARS['txtphone'];
$email = $HTTP_POST_VARS['txtemail'];
$major = $HTTP_POST_VARS['txtmajor'];
$enrollment = $HTTP_POST_VARS['txtenrollment'];

$msg = "Student ID: \t$studentid\n";
$msg .= "Last Name: \t$lastname\n";
$msg .= "First Name: \t$firstname\n";
$msg .= "Gender: \t$gender\n";
$msg .= "Phone: \t$phone\n";
$msg .= "Email: \t$email\n";
$msg .= "Major: \t$major\n";
$msg .= "Enrollment: \t$enrollment\n";

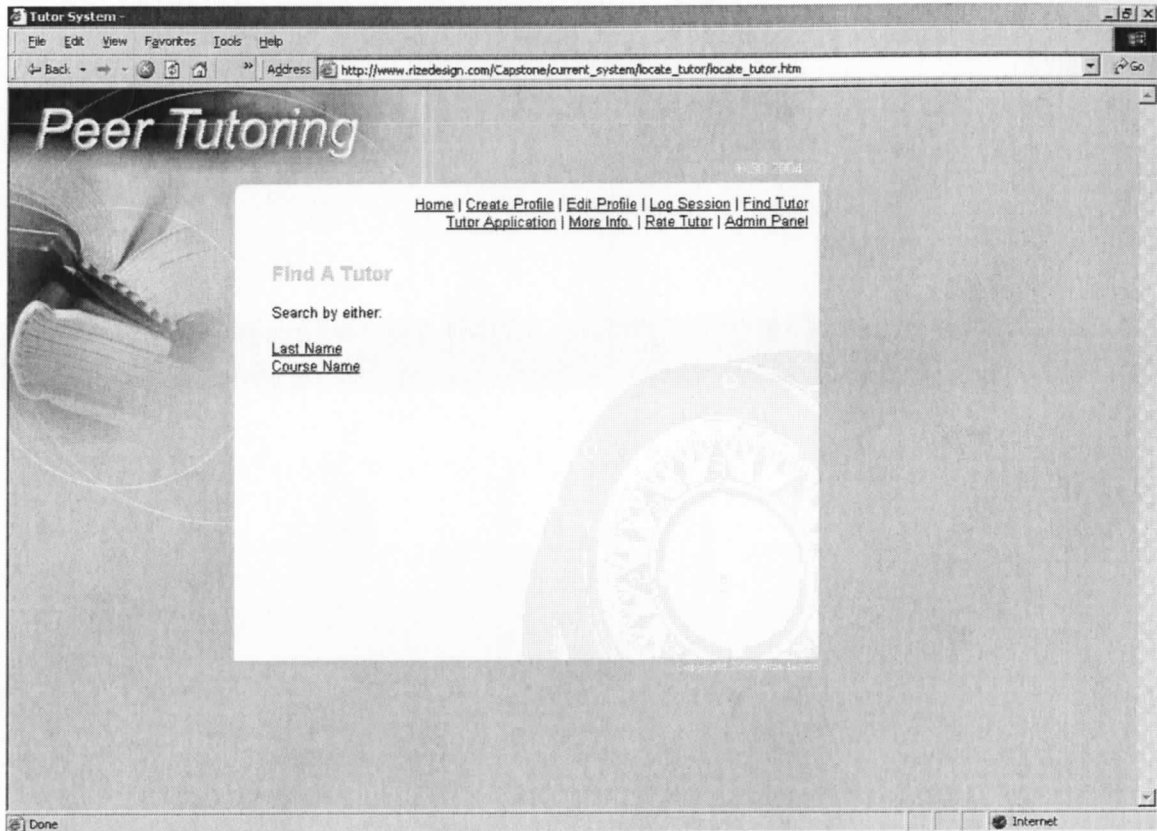
mail("test@test.com", "Feedback Form", $msg, $mailheaders);?>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr>
```

```

<td align="left" valign="top">Thank you, the Dean will contact you shortly. </td>
<td></td>
</tr>
</table>
<!-- #EndEditable -->

```

locate_tutor.htm



```

<!-- #BeginEditable "main_content" -->
<table width="350" border="0" cellspacing="0" cellpadding="8">
<tr>
<td>
<p><b><font color="#CCCCCC" size="4">Find A Tutor</font></b></p>
</td>
</tr>
<tr>
<td>Search by either:</td>
</tr>
<tr>
<td align="left" valign="top"> <a href="locate_tutor_name.htm">Last
Name</a><br>
<a href="locate_tutor_course.php">Course Name</a><br>
<p>&nbsp;</p>
</td>

```

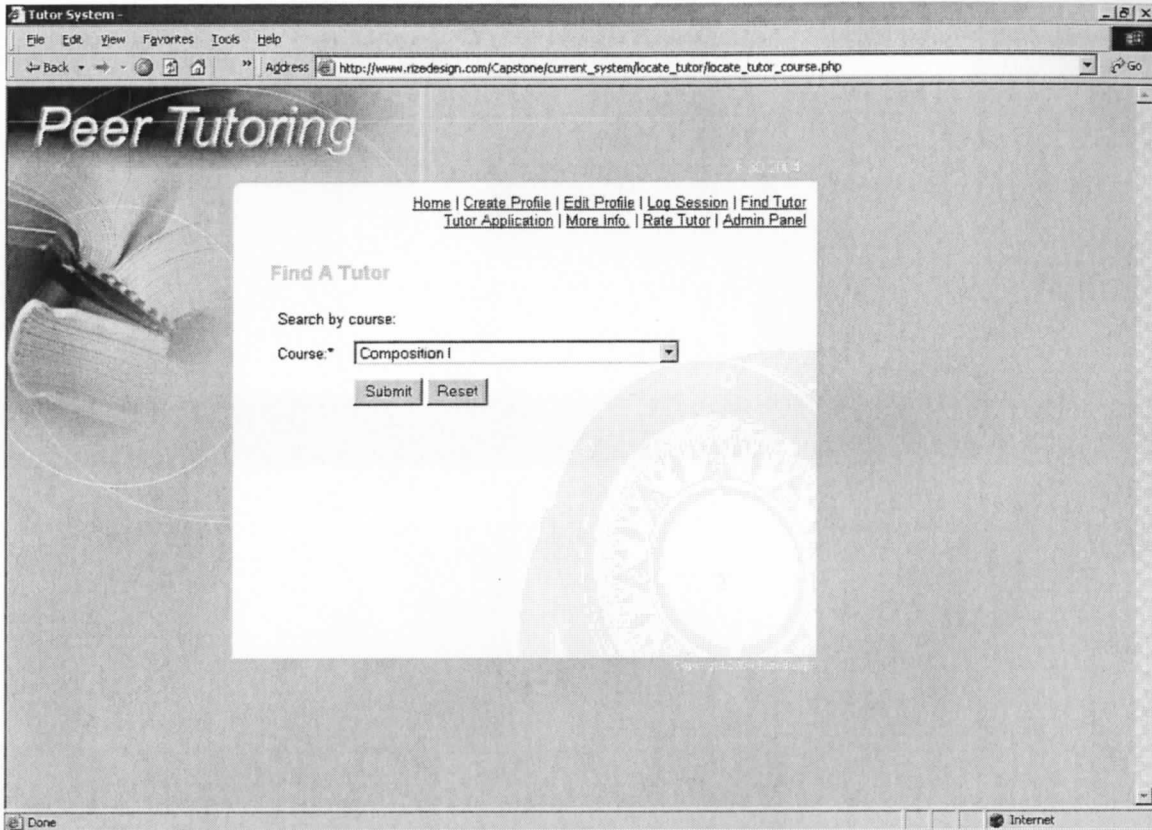


```

</tr>
</table>
<!-- #EndEditable -->

```

locate_tutor_course.php



```

<!-- #BeginEditable "main_content" -->
<table width="350" border="0" cellspacing="0" cellpadding="8">
<tr>
<td>
<p><b><font color="#CCCCCC" size="4">Find A Tutor</font></b></p>
</td>
</tr>
<tr>
<td align="left" valign="top">
<form name="form1" method="post" action="query_tutor_course.php">
<table width="400" border="0" cellpadding="5">
<tr align="left" valign="top">
<td colspan="2">
<p>Search by course: </p>
</td>
</tr>
<tr>
<td>Course:*

```



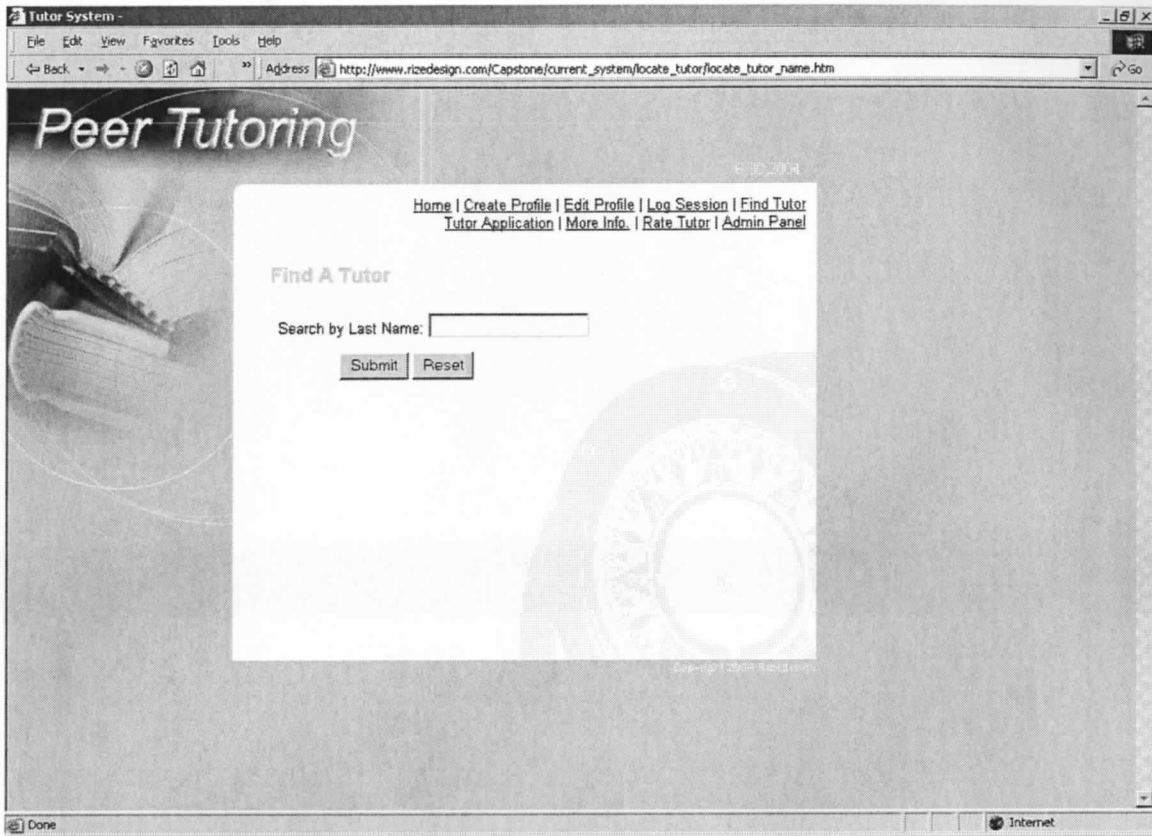
```

<td> <font size="2">
<SELECT name="course">
<?php

include("../includes/db.php");
MYSQL_CONNECT(HOST,USER,PASS) OR DIE("Unable to connect to database");
@mysql_select_db(DB) or die("Unable to select database");
$query="select * from course";
$result=mysql_query($query) or die("Unable to Make the Query:" . mysql_error() );
while($row=mysql_fetch_array($result)){
echo "<OPTION VALUE=".$row['Course_ID'].>".$row['Course_Name'].</OPTION>";
}
?>
</select>
</font></td>
</tr>
<tr>
<td>&nbsp; </td>
<td>
<input type="submit" name="Submit" value="Submit">
<input type="reset" name="Submit2" value="Reset">
</td>
</tr>
</table>
</form>
<p>&nbsp;</p>
</td>
</tr>
</table>
<!-- #EndEditable -->

```

locate_tutor_name.php



```

<!-- #BeginEditable "main_content" -->
<table width="350" border="0" cellspacing="0" cellpadding="8">
<tr>
<td>
<p><b><font color="#CCCCCC" size="4">Find A Tutor</font></b></p>
</td>
</tr>
<tr>
<td align="left" valign="top">
<form name="form1" method="post" action="query_tutor.php">
<table width="400" border="0" cellpadding="5">
<tr align="left" valign="top">
<td colspan="2">
<p>Search by Last Name:<font size="2"> </font>
<input type="text" name="lastname">
</td>
</tr>
<tr>
<td>&nbsp;   </td>
<td>
<input type="submit" name="Submit" value="Submit">
<input type="reset" name="Submit2" value="Reset">

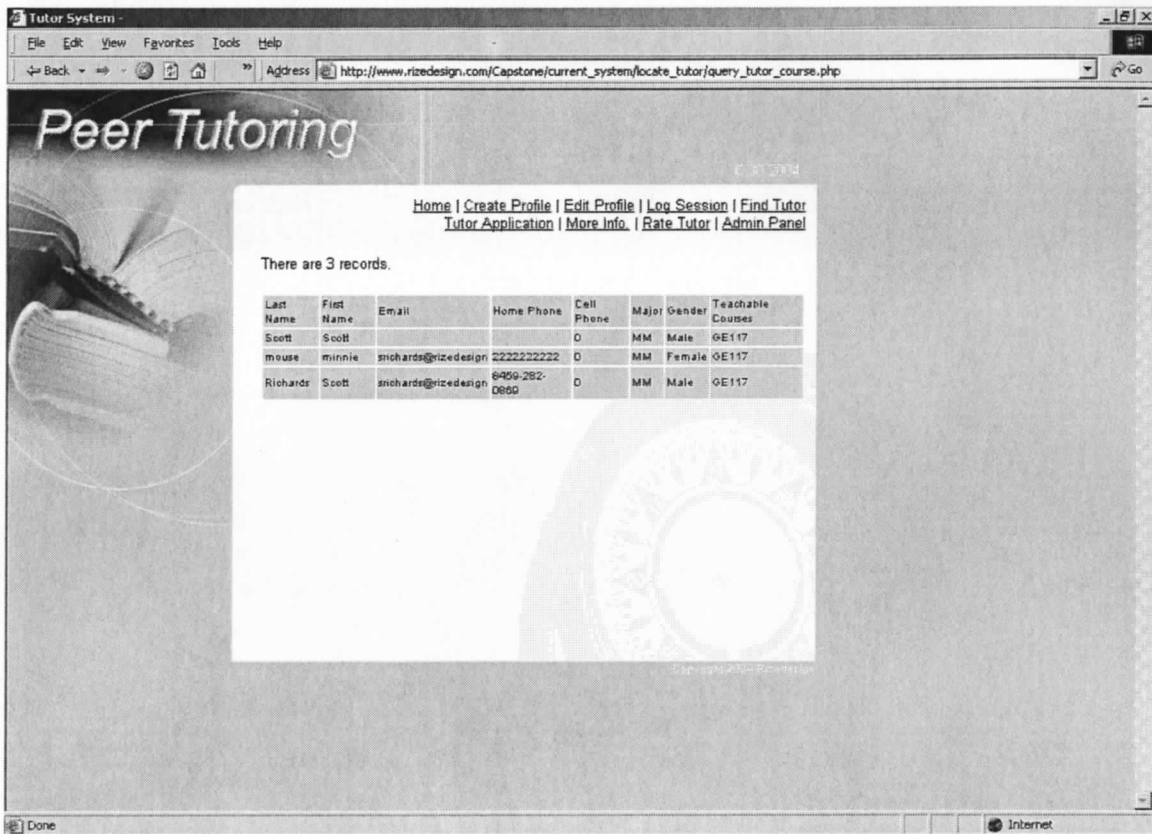
```

```

</td>
</tr>
</table>
</form>
<p>&nbsp;</p>
</td>
</tr>
</table>
<!-- #EndEditable -->

```

query_tutor_course.php



```

<!-- #BeginEditable "main_content" -->
<?php
$coursename = $_HTTP_POST_VARS['coursename'];
$db="wyatt1_tutorsystem";
$link = mysql_connect("localhost", "wyatt1_itt", "student");
if (! $link)
die("Couldn't connect to MySQL");
mysql_select_db($db, $link)
or die("Couldn't open $db: ".mysql_error());
$result = mysql_query(
"SELECT
student_information.Last_Name,student_information.First_Name,student_information.Email

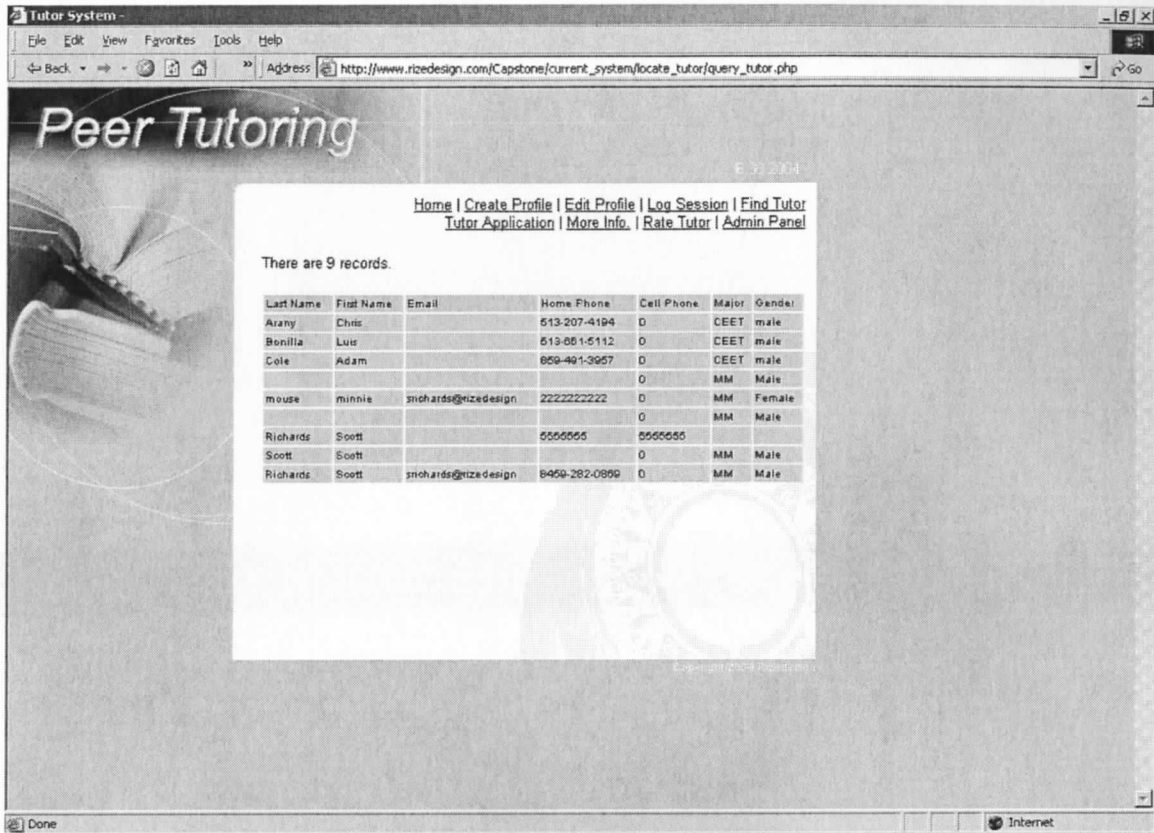
```

```

,student_information.Home_Phone,student_information.Cell_Phone,student_information.Ma
jor,student_information.Gender, teachable_courses.Course_Name
FROM student_information,teachable_courses WHERE
student_information.disposition='tutor' AND teachable_courses.Course_Name
=' $coursename' AND student_information.student_id = teachable_courses.student_id")
or die("SELECT Error: ".mysql_error());
$num_rows = mysql_num_rows($result);
print "There are $num_rows records.<P>";
print "<table width='100%'\><b><tr bgcolor=' #cccc99'\><td><font face=arial
size=1/>Last Name</font></td><td><font face=arial size=1/>First
Name</font></td><td><font face=arial size=1/>Email</font></td><td><font face=arial
size=1/>Home Phone</font></td><td><font face=arial size=1/>Cell
Phone</font></td><td><font face=arial size=1/>Major</font></td><td><font face=arial
size=1/>Gender</font></td><td><font face=arial size=1/>Teachable
Courses</font></td></tr></b>";
while ($get_info = mysql_fetch_row($result)){
print "<tr bgcolor=' #cccccc'\>\n";
foreach ($get_info as $field)
print "\t<td><font face=arial size=1/>$field</font></td>\n";
print "</tr>\n";
}
print "</table>\n";
mysql_close($link);
?>
<!-- #EndEditable -->

```

query_tutor.php



```
<!-- #BeginEditable "main_content" -->
```

```
<?php
```

```
$lastname = $HTTP_POST_VARS['lastname'];
```

```
$db="wyatt1_tutorsystem";
```

```
$link = mysql_connect("localhost", "wyatt1_itt", "student");
```

```
if (! $link)
```

```
die("Couldn't connect to MySQL");
```

```
mysql_select_db($db, $link)
```

```
or die("Couldn't open $db: ".mysql_error());
```

```
$result = mysql_query( "SELECT
```

```
student_information.Last_Name,student_information.First_Name,
```

```
student_information.Email,student_information.Home_Phone,student_information.Cell_Pho
```

```
ne,student_information.Major,student_information.Gender
```

```
FROM student_information
```

```
WHERE Last_Name
```

```
LIKE '$lastname%'
```

```
AND disposition = 'tutor'")
```

```
or die("SELECT Error: ".mysql_error());
```

```
$num_rows = mysql_num_rows($result);
```

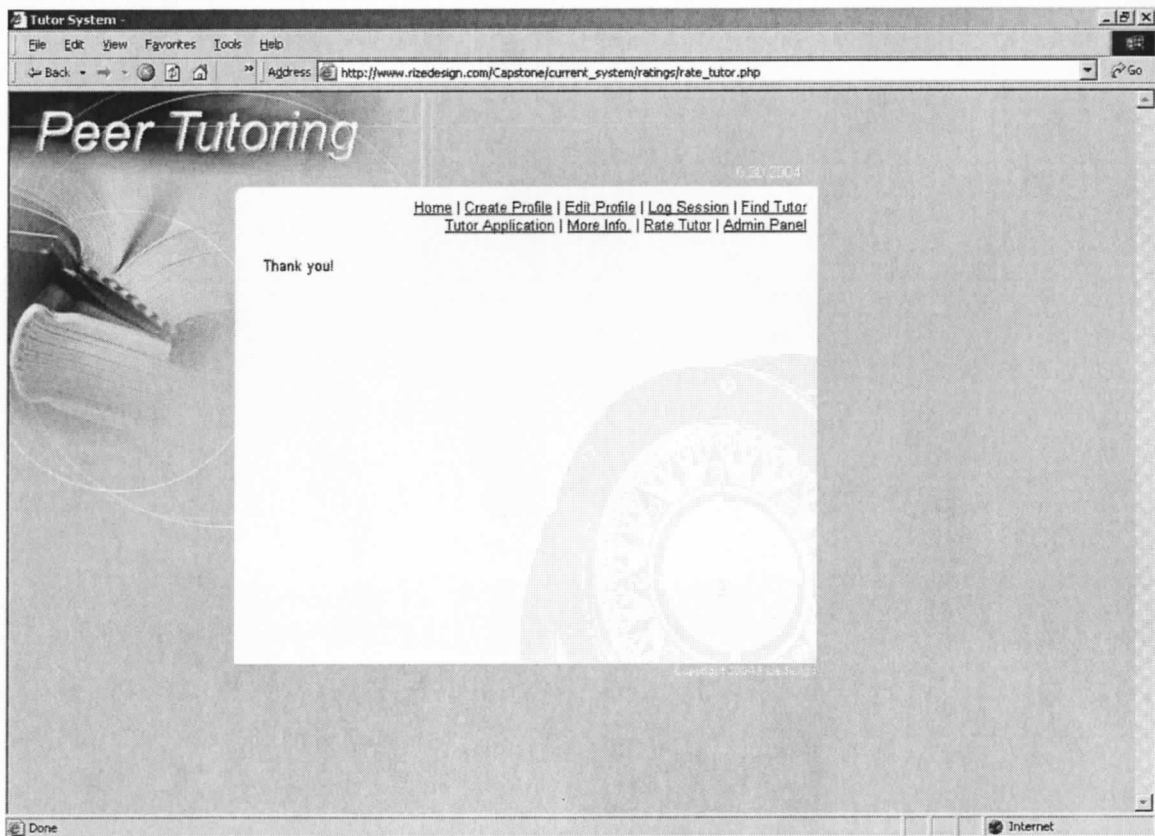
```
print "There are $num_rows records.<P>";
```

```

echo "<table width='\"100%\"'><b><tr bgcolor='\"#cccc99\"'><td><font face=arial
size=1/>Last Name</font></td><td><font face=arial size=1/>First
Name</font></td><td><font face=arial size=1/>Email</font></td><td><font face=arial
size=1/>Home Phone</font></td><td><font face=arial size=1/>Cell
Phone</font></td><td><font face=arial size=1/>Major</font></td><td><font face=arial
size=1/>Gender</font></td></tr></b>";
while ($get_info = mysql_fetch_row($result)){
print "<tr bgcolor='\"#cccccc\"'>\n";
foreach ($get_info as $field)
print "\t<td><font face=arial size=1/>$field</font></td>\n";
print "</tr>\n";
}
print "</table>\n";
mysql_close($link);
?>
<!-- #EndEditable -->

```

rate_tutor.php



```

<!-- #BeginEditable "main_content" -->
<?php
$tutor = $HTTP_POST_VARS['tutor'];
$tutee = $HTTP_POST_VARS['tutee'];
$course = $HTTP_POST_VARS['course']

```

```

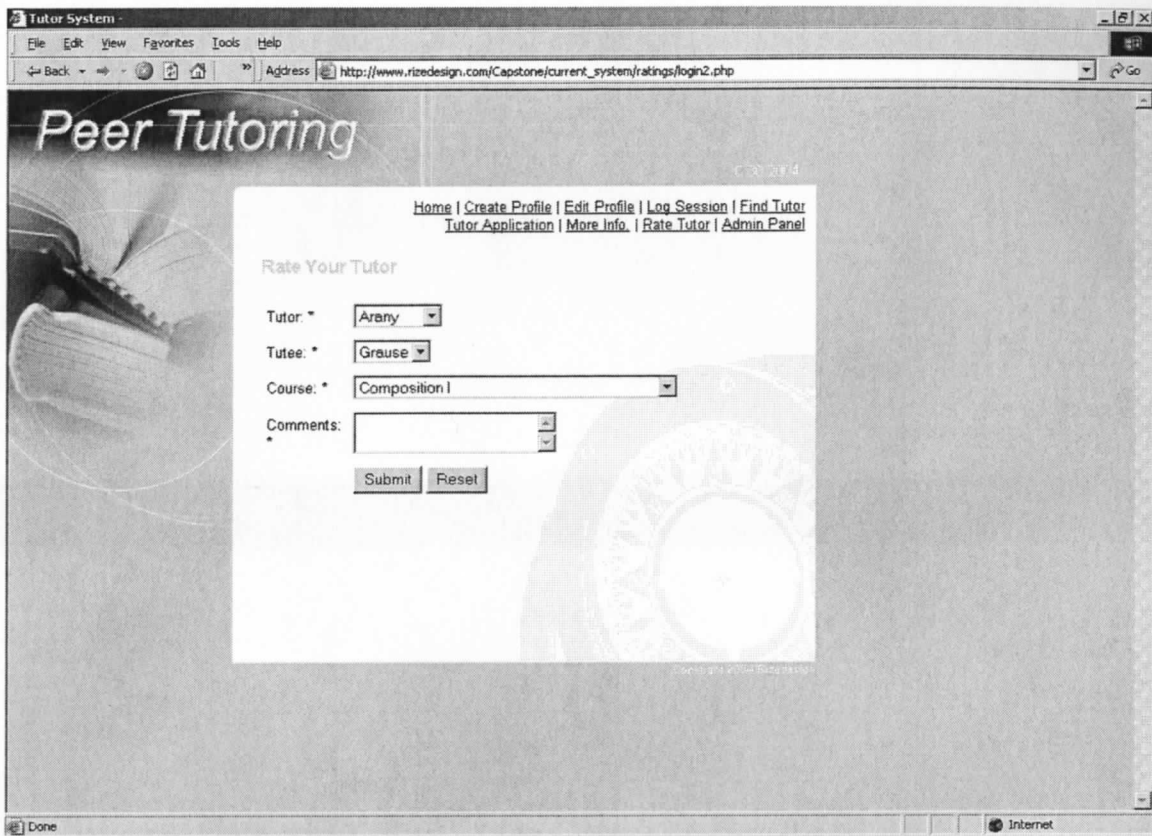
$comments = $HTTP_POST_VARS['comments'];

$db="wyatt1_tutorsystem";
$link = mysql_connect("localhost", "wyatt1_itt", "student");
if (! $link)
die("Couldn't connect to MySQL");
mysql_select_db($db , $link)
or die("Couldn't open $db: ".mysql_error());
if (!$tutor && !$tutee && !$course && !$comments)
{
    echo "you didn't fill in all the required fields! go back and do it again...";
}

else
{
$query = "INSERT INTO rate_tutor (tutor,tutee,course,comments) VALUES
('$tutor','$tutee','$course','$comments')";
$result = mysql_query($query, $link);
echo "Thank you!";
}
session_destroy();
?>
<!-- #EndEditable -->

```


rate_tutor_form.php



```
<!-- #BeginEditable "main_content" -->
<p><b><font color="#CCCCCC" size="3">Rate Your Tutor</font></b></p>
<form method="post" action="rate_tutor.php">
<table width="300" border="0" cellspacing="0" cellpadding="5">
<tr>
<td>Tutor: *</td>
<td>
<select name="tutor">
<?php include("../includes/db.php");

MYSQL_CONNECT(HOST,USER,PASS) OR DIE("Unable to connect to database");
@mysql_select_db(DB) or die("Unable to select database");
$query=("select Last_Name from student_information WHERE disposition LIKE 'tutor'");
$result=mysql_query($query) or die("Unable to Make the Query:" . mysql_error() );
while($row=mysql_fetch_array($result)){
echo "<OPTION VALUE=".$row['Last_Name'].>".$row['Last_Name'].</OPTION>";
}

?>
</select>
```



```

</td>
</tr>
<tr>
<td>Tutee: *</td>
<td> <select name="tutee"><?php

include("../includes/db.php");
MYSQL_CONNECT(HOST,USER,PASS) OR DIE("Unable to connect to database");
@mysql_select_db(DB) or die( "Unable to select database");
$query=("select Last_Name from student_information WHERE disposition LIKE 'tutee'");
$result=mysql_query($query) or die ("Unable to Make the Query:" . mysql_error() );
while($row=mysql_fetch_array($result)){
echo "<OPTION VALUE=".$row['Last_Name'].>".$row['Last_Name'].</OPTION>";
}

?> </select></td>
</tr>
<tr>
<td>Course: *</td>
<td>
<SELECT name="course">
<?php include("../includes/db.php");

MYSQL_CONNECT(HOST,USER,PASS) OR DIE("Unable to connect to database");
@mysql_select_db(DB) or die( "Unable to select database");
$query=("select * from course");
$result=mysql_query($query) or die ("Unable to Make the Query:" . mysql_error() );
while($row=mysql_fetch_array($result)){
echo "<OPTION VALUE=".$row['Course_ID'].>".$row['Course_Name'].</OPTION>";
}
?>
</select>
</td>
</tr>
<tr>
<td>Comments: *</td>
<td>
<textarea name="comments"></textarea>
</td>
</tr>
<tr>
<td>&nbsp;</td>
<td>
<input type="submit" name="Submit" value="Submit">
<input type="reset" name="Submit2" value="Reset">
</td>

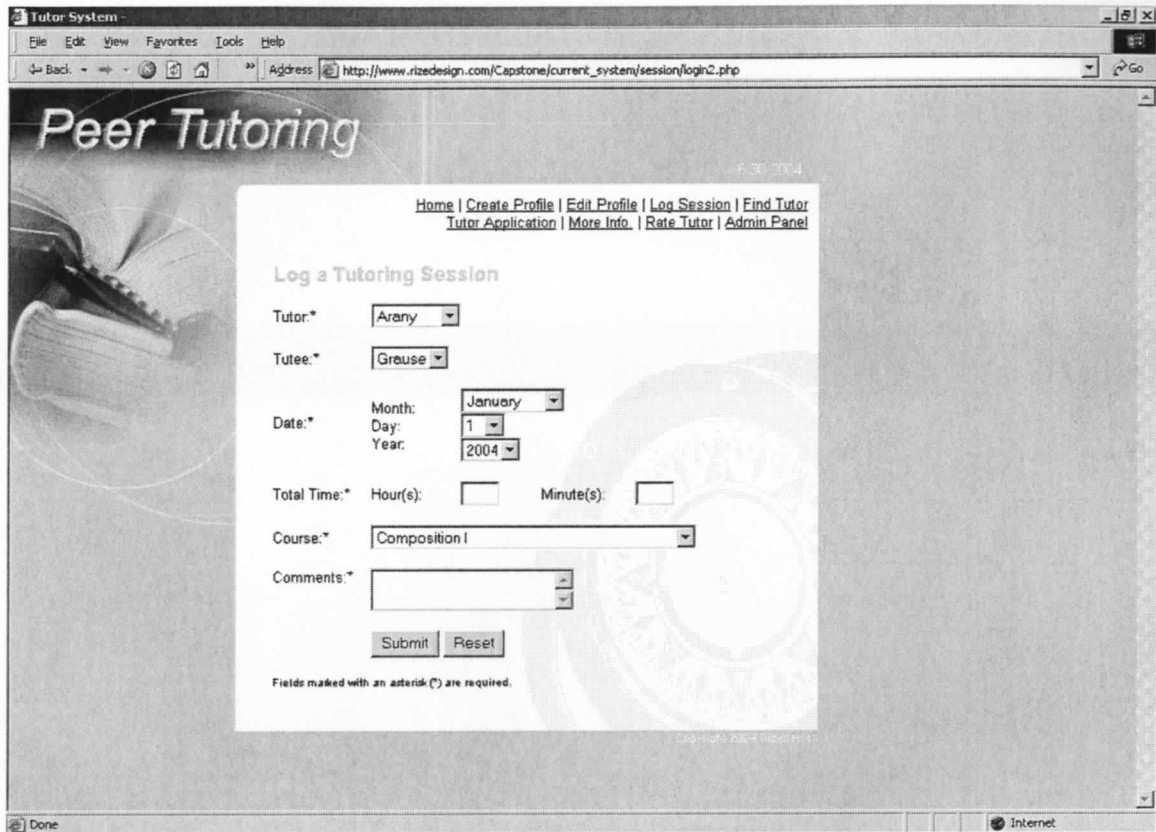
```

```

</tr>
</table>
</form>
<p>&nbsp;</p>
<!-- #EndEditable -->

```

session.php



```

<!-- #BeginEditable "main_content" -->
<form name="form1" method="post" action="log_tutor_session.php">
<table width="351" border="0" cellspacing="0" cellpadding="8">
<tr>
<td colspan="5">
<p><b><font color="#CCCCCC" size="4">Log a Tutoring
Session </font></b></p>
</td>
</tr>
<tr>
<td>Tutor:*</td>
<td colspan="4">
<select name="tutor">
<?php

```

```

include("../includes/db.php");

```

```

MYSQL_CONNECT(HOST,USER,PASS) OR DIE("Unable to connect to database");
@mysql_select_db(DB) or die( "Unable to select database");
$query=("select Last_Name from student_information WHERE disposition LIKE 'tutor'");
$result=mysql_query($query) or die ("Unable to Make the Query:" . mysql_error() );
while($row=mysql_fetch_array($result)){
echo "<OPTION VALUE=".$row['Last_Name']. ">".$row['Last_Name']. "</OPTION>";
}
?>
</select>
</td>
</tr>
<tr>
<td>Tutee:*</td>
<td colspan="4">
<select name="tutee">
<?php

```

```

include("../includes/db.php");
MYSQL_CONNECT(HOST,USER,PASS) OR DIE("Unable to connect to database");
@mysql_select_db(DB) or die( "Unable to select database");
$query=("select Last_Name from student_information WHERE disposition LIKE 'tutee'");
$result=mysql_query($query) or die ("Unable to Make the Query:" . mysql_error() );
while($row=mysql_fetch_array($result)){
echo "<OPTION VALUE=".$row['Last_Name']. ">".$row['Last_Name']. "</OPTION>";
}
?>
</select>
</td>
</tr>
<tr>
<td>Date:*</td>
<td rowspan="3"> Month: <br>
Day: <br>
Year: </td>
<td colspan="3" rowspan="3"><font size="2">
<select name="month">
<option selected>January</option>
<option>February</option>
<option>March</option>
<option>April</option>
<option>May</option>
<option>June</option>
<option>July</option>
<option>August</option>
<option>September</option>
<option>October</option>

```

```
<option>November</option>
<option>December</option>
</select>
<br>
</font><font size="2">
<select name="day">
<option selected>1</option>
<option>2</option>
<option>3</option>
<option>4</option>
<option>5</option>
<option>6</option>
<option>7</option>
<option>8</option>
<option>9</option>
<option>10</option>
<option>11</option>
<option>12</option>
<option>13</option>
<option>14</option>
<option>15</option>
<option>16</option>
<option>17</option>
<option>18</option>
<option>19</option>
<option>20</option>
<option>21</option>
<option>22</option>
<option>23</option>
<option>24</option>
<option>25</option>
<option>26</option>
<option>27</option>
<option>28</option>
<option>29</option>
<option>30</option>
<option>31</option>
</select>
</font><font size="2"> <br>
<select name="year">
<option selected>2004</option>
<option>2005</option>
<option>2006</option>
<option>2007</option>
<option>2008</option>
<option>2009</option>
```

```

<option>2010</option>
</select>
</font></td>
</tr>
<tr>
<td><font size="2"></font></td>
</tr>
<tr>
<td><font size="2"></font></td>
</tr>
<tr>
<td>Total Time:* </td>
<td> Hour(s): </td>
<td><font size="2">
<input type="text" name="hour" size="2">
</font></td>
<td>Minute(s): </td>
<td><font size="2">
<input type="text" name="minute" size="2">
</font></td>
</tr>
<tr>
<td>Course:*
<td colspan="4">
<SELECT name="course_name">
<?php

```

```

include("../includes/db.php");
MYSQL_CONNECT(HOST,USER,PASS) OR DIE("Unable to connect to database");
@mysql_select_db(DB) or die("Unable to select database");
$query="select * from course";
$result=mysql_query($query) or die("Unable to Make the Query:" . mysql_error() );
while($row=mysql_fetch_array($result)){
echo "<OPTION VALUE=".$row['Course_ID'].>".$row['Course_Name'].</OPTION>";
}
?>
</select>
</td>
</tr>
<tr>
<td align="left" valign="top">Comments:*</td>
<td colspan="4">
<textarea name="comments"></textarea>
</td>
</tr>
<tr>

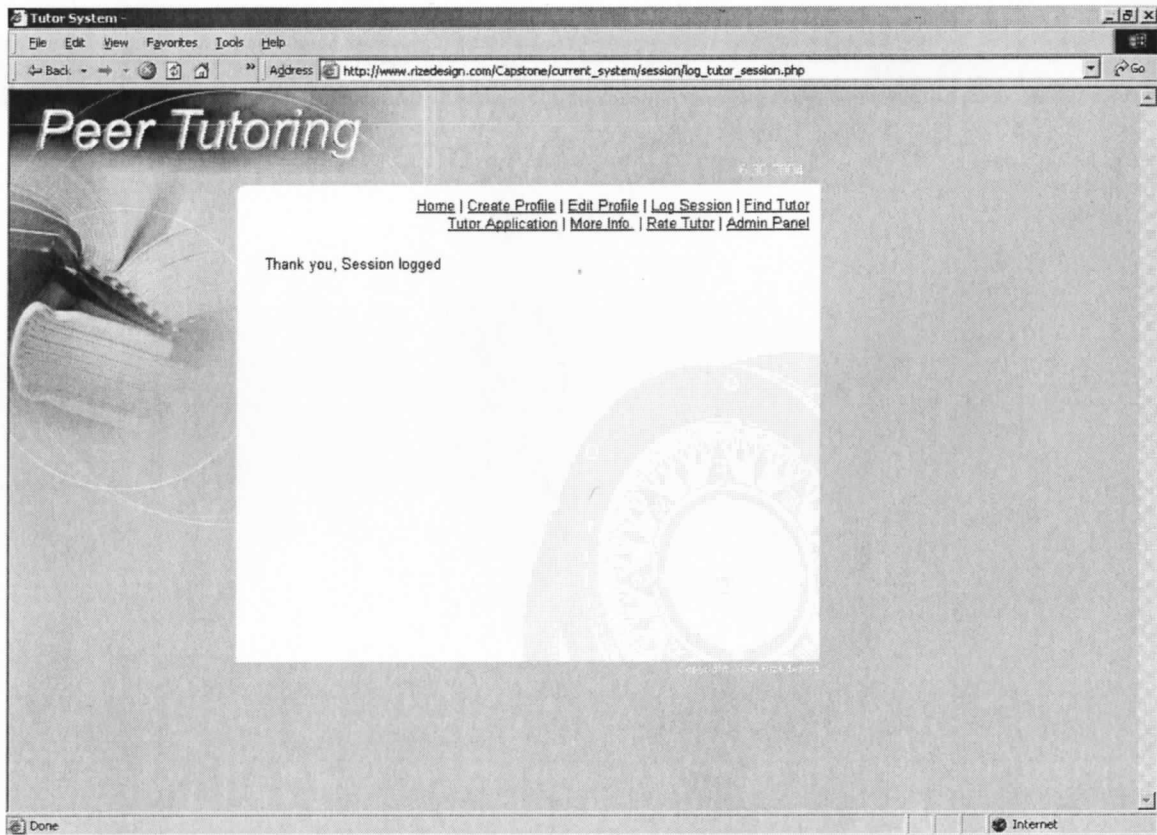
```

```

<td>&nbsp;</td>
<td colspan="4">
<input type="submit" name="Submit" value="Submit">
<input type="reset" name="Submit2" value="Reset">
</td>
</tr>
<tr>
<td colspan="5"><font size="1">Fields marked with an asterisk (*) are
required.</font></td>
</tr>
</table>
</form>
<!-- #EndEditable -->

```

log_tutor_session.php



```

<!-- #BeginEditable "main_content" -->
<?php

$tutor = $HTTP_POST_VARS['tutor'];
$tutee = $HTTP_POST_VARS['tutee'];

$hour = $HTTP_POST_VARS['hour'];
$minute = $HTTP_POST_VARS['minute'];

```

```

$comments = $HTTP_POST_VARS['comments'];
$course_name = $HTTP_POST_VARS['course_name'];
$day = $HTTP_POST_VARS['day'];
$month = $HTTP_POST_VARS['month'];
$year = $HTTP_POST_VARS['year'];

$db="wyatt1_tutorsystem";
$link = mysql_connect("localhost", "wyatt1_itt", "student");
if (! $link)
die("Couldn't connect to MySQL");
mysql_select_db($db , $link)
or die("Couldn't open $db: ".mysql_error());

if (!$tutor && !$tutee && !$hour && !$minute && !$comments && !$course_name &&
!$day && !$month && !$year)
{
echo "you didn't fill in all the required fields! go back and do it again...";
}
else
{
$query = "INSERT INTO tutor_session (hour, minute ,day, month,
year,tutee,tutor,comments,course) VALUES
('$hour', '$minute', '$day', '$month', '$year', '$tutee', '$tutor', '$comments', '$course_name')";
$result = mysql_query($query, $link);
echo "Thank you, Session logged";
}
session_destroy();
?>
<!-- #EndEditable -->

```

login.htm

The screenshot shows a Microsoft Internet Explorer window titled "Tutor System - Microsoft Internet Explorer". The address bar displays "http://www.rizedesign.com/Capstone/current_system/edk_profile/tutor/login.htm". The page content includes a "Peer Tutoring" header, a navigation menu with links like "Home", "Create Profile", "Edit Profile", "Log Session", "Find Tutor", "Tutor Application", "More Info", "Rate Tutor", and "Admin Panel", and a "Please Log In" form. The form contains three input fields: "Username:*", "Paseword:*" (note the typo), and "Student ID:*". Below the fields are "Submit" and "Reset" buttons. A note at the bottom of the form states "Fields marked with an asterisk (*) are required." The background of the page features a faint image of a hand holding a pen over a document.

```
<!--#BeginEditable "main_content" -->
<form name="form1" method="post" action="login2.php">
<table width="351" border="0" cellspacing="0" cellpadding="8">
<tr>
<td colspan="2">
<p><b><font color="#CCCCCC" size="4">Please Log In</font></b></p>
</td>
</tr>
<tr>
<td>Username:<!--#BeginEditable "main_content" -->
<td>
<input type="text" name="username" >
</td>
</tr>
<tr>
<td>Paseword:<!--#BeginEditable "main_content" -->
<td>
<input type="password" name="password" >
</td>
</tr>
</table>
```

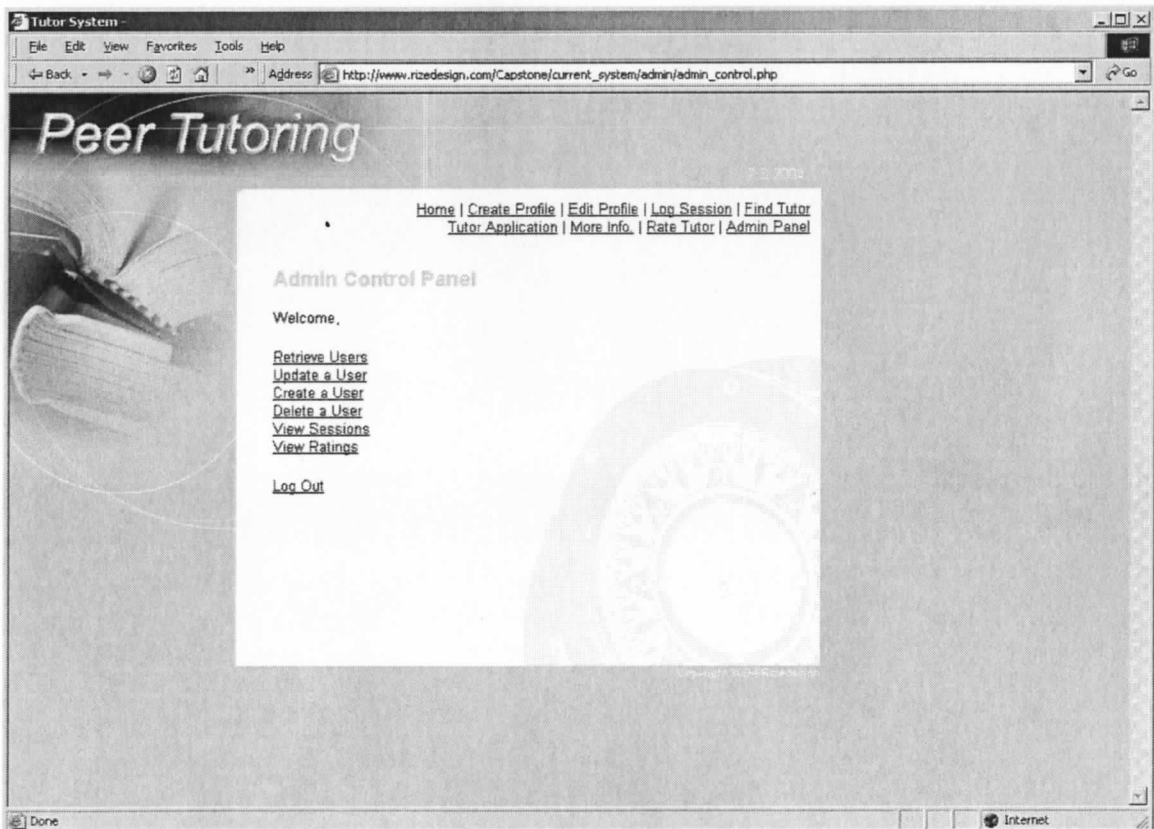


```

<tr>
<td>Student ID:*</td>
<td>
<input type="text" name="studentid" >
</td>
</tr>
<tr>
<td colspan="2">
<input type="submit" name="Submit" value="Submit">
<input type="reset" name="Submit2" value="Reset">
</td>
</tr>
<tr>
<td colspan="2"><font size="1">Fields marked with an asterisk (*) are
required.</font></td>
</tr>
</table>
</form>
<!-- #EndEditable -->

```

Admin_control.php



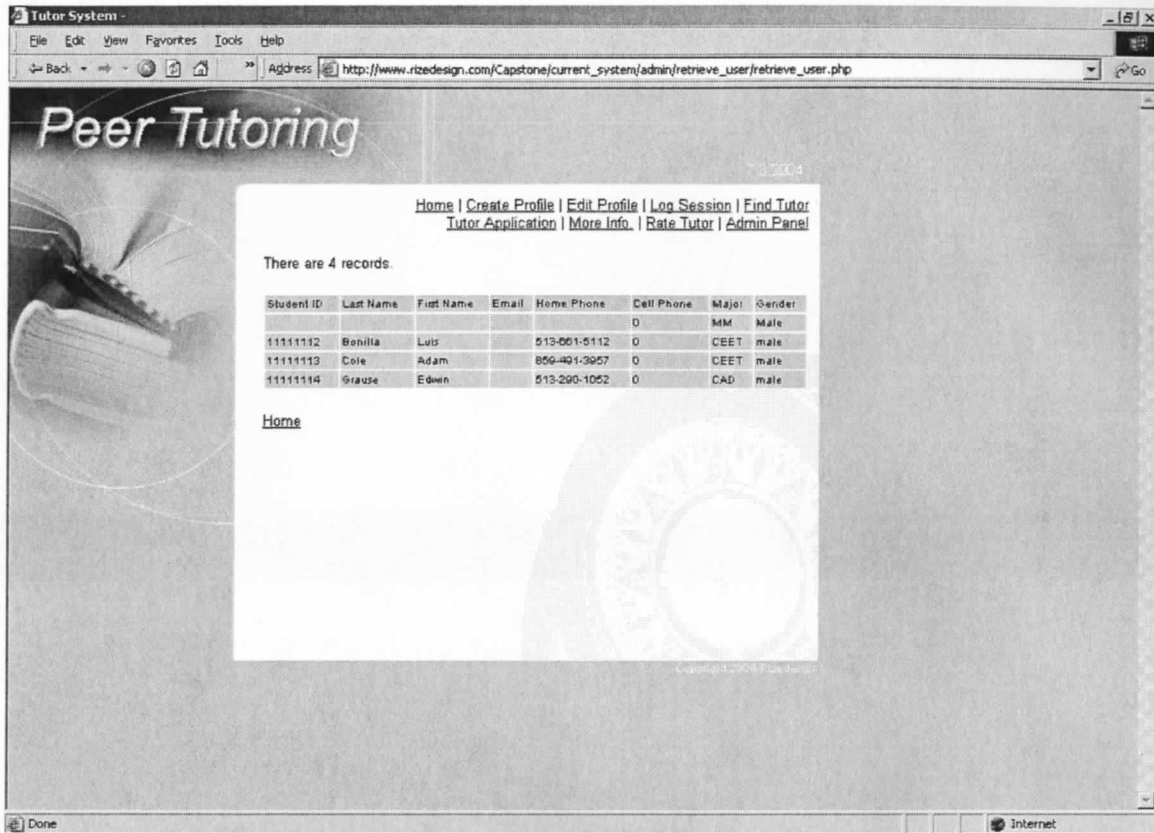
```

<!-- #BeginEditable "main_content" -->
<table width="348" border="0" cellspacing="0" cellpadding="8">

```

```
<tr>
<td colspan="2">
<p><b><font size="4" color="#CCCCCC">Admin Control
Panel</font></b></p>
</td>
</tr>
<tr>
<td colspan="2">
<p>Welcome<font size="2">,</font></p>
<p><a href="retrieve_user/retrieve_user.php">Retrieve
User</a><br>
<a href="update_user.php"> Update a User</a><br>
<a href="add_user/create_profile.htm">Create a
User</a><br>
<a href="delete_user/delete_user.php"> Delete
a User </a><br>
<a href="sessions/view_sessions.php">View Sessions</a><br>
<a href="ratings/view_ratings.php">View Ratings</a></p>
<p><a href="kill_session.php">Log Out</a></p>
</td>
</p>
</td>
</tr>
</table>
<p>&nbsp;</p>
<!-- #EndEditable -->
```

Retrieve_user.php



```
<!-- #BeginEditable "main_content" -->
<?php
```

```
$db="wyatt1_tutorsystem";
$link = mysql_connect("localhost", "wyatt1_itt", "student");
if (!$link)
die("Couldn't connect to MySQL");
mysql_select_db($db, $link)
or die("Couldn't open $db: ".mysql_error());
```

```
$result = mysql_query(
"SELECT
student_information.student_id,student_information.Last_Name,student_information.First_N
ame,student_information.Email,student_information.Home_Phone,student_information.Cell
Phone,student_information.Major,student_information.Gender
FROM student_information")
or die("SELECT Error: ".mysql_error());
```

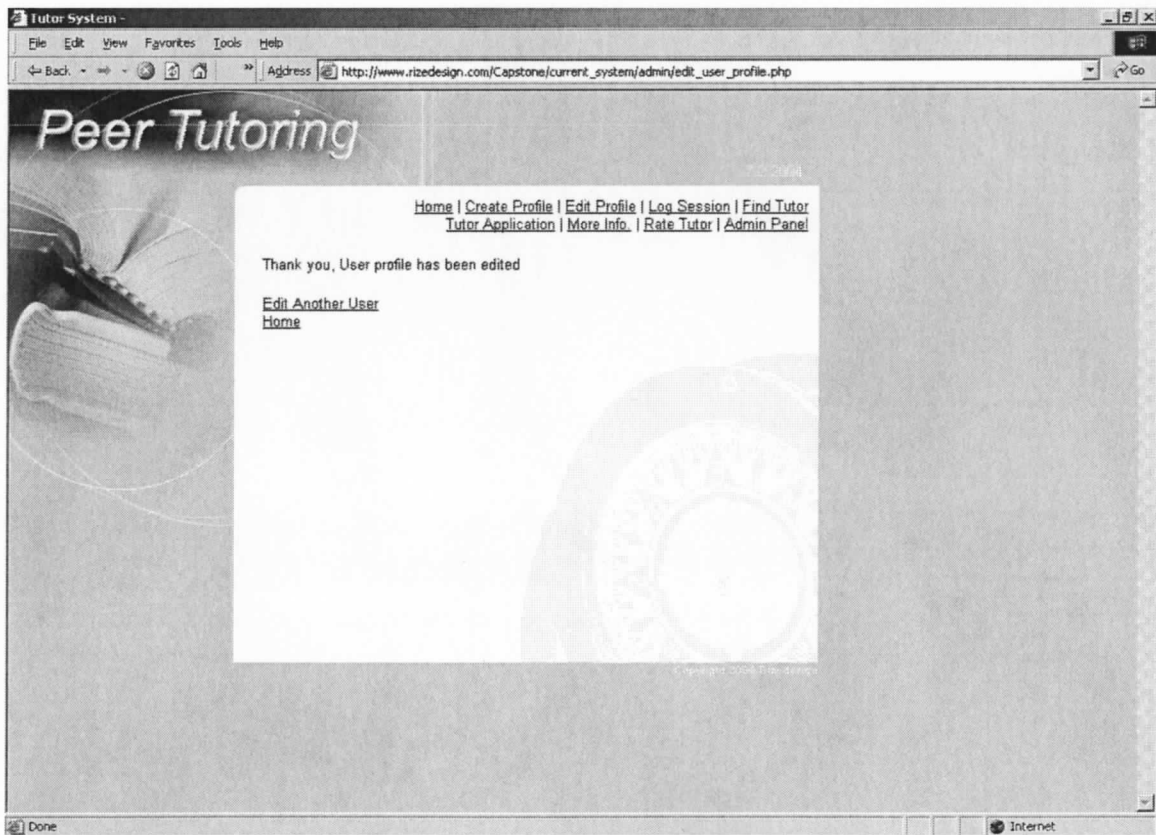
```
$num_rows = mysql_num_rows($result);
print "There are $num_rows records.<P>";
```

```

echo "<table width='100%'><b><tr bgcolor='#cccc99'><td><font face=arial
size=1/>Student ID</font></td><td><font face=arial size=1/>Last
Name</font></td><td><font face=arial size=1/>First Name</font></td><td><font
face=arial size=1/>Email</font></td><td><font face=arial size=1/>Home
Phone</font></td><td><font face=arial size=1/>Cell Phone</font></td><td><font
face=arial size=1/>Major</font></td><td><font face=arial
size=1/>Gender</font></td></tr></b>";
while ($get_info = mysql_fetch_row($result)){
print "<tr bgcolor='#cccccc'>\n";
foreach ($get_info as $field)
print "\t<td><font face=arial size=1/>$field</font></td>\n";
print "</tr>\n";
}
print "</table>\n";
mysql_close($link);
?>
<p><a href='../admin_control.php'>Home</a></p>
<!-- #EndEditable -->

```

edit_user_profile.php



```

<!-- #BeginEditable "main_content" --> <?php
$studentid = $HTTP_POST_VARS['studentid'];
$firstname = $HTTP_POST_VARS['firstname'];

```

```

$lastname = $HTTP_POST_VARS['lastname'];
$phone = $HTTP_POST_VARS['phone'];
$email = $HTTP_POST_VARS['email'];
$gender = $HTTP_POST_VARS['gender'];
$major = $HTTP_POST_VARS['major'];
$availability = $HTTP_POST_VARS['availability'];
$username = $HTTP_POST_VARS['username'];
$password = $HTTP_POST_VARS['password'];
$coursename = implode(',', $_POST['coursename']);

$db="wyatt1_tutorsystem";
$link = mysql_connect("localhost", "wyatt1_itt", "student");
if (! $link)
die("Couldn't connect to MySQL");
mysql_select_db($db, $link)
or die("Couldn't open $db: ".mysql_error());

if (!$studentid && !$firstname && !$lastname && !$phone && !$email && !$gender &&
!$major && !$disposition && !$availability)
{
echo "you didn't fill in all the required fields! go back and do it again...";
}

else
{

columns.
$query = "UPDATE student_information
SET
student_id='$studentid',First_Name='$firstname',Last_Name='$lastname',Home_Phone='$
phone',email='$email',Gender='$gender',Major='$major',availability='$availability'
WHERE Student_ID = '$studentid'";
$result = mysql_query($query, $link);

$query2 ="UPDATE login SET
student_ID='$studentid',username='$username',password='$password' WHERE
student_id='$studentid'";
$result = mysql_query($query2, $link);
$sql = "UPDATE teachable_courses SET
student_ID='$studentid',Course_Name='$coursename' WHERE Student_ID='$studentid'";
$result = mysql_query($sql);
echo "Thank you, User profile has been edited";
}
session_destroy();
?>

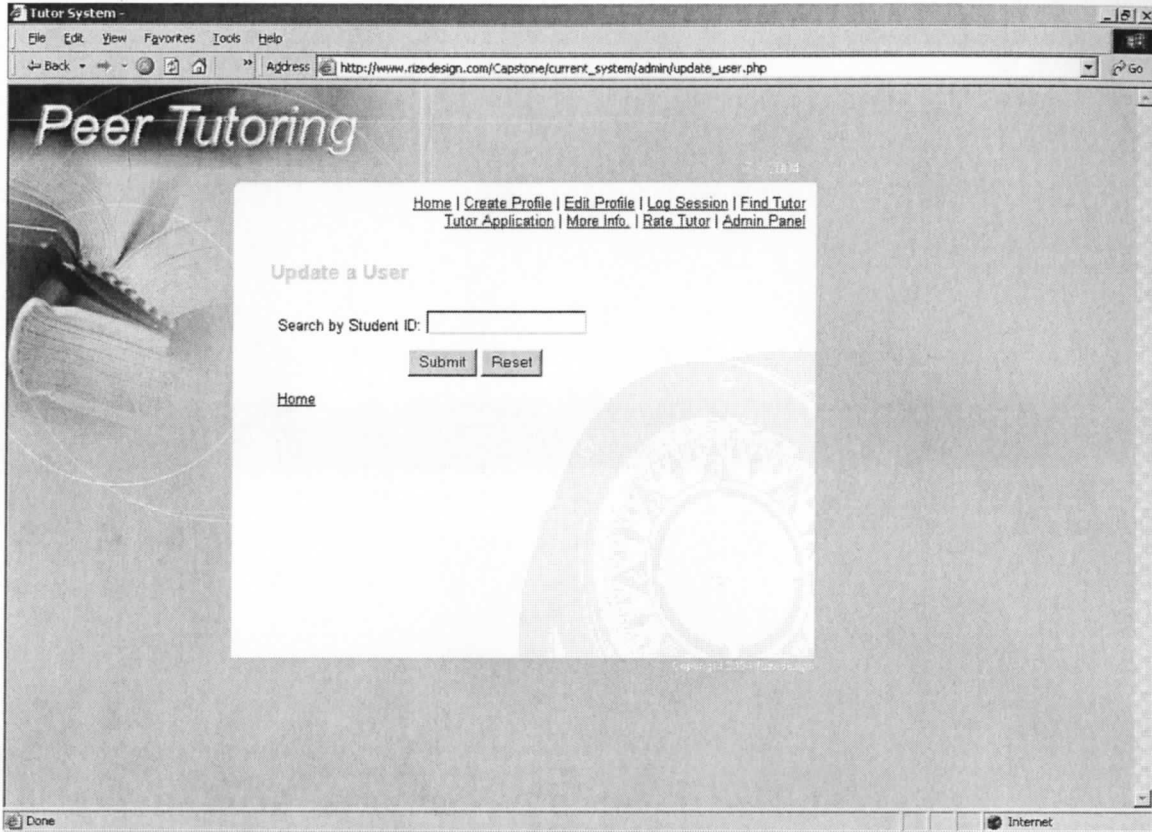
```

```

<p><a href="update_user.php">Edit Another User<br>
</a><a href="admin_control.php">Home</a>
<!-- #EndEditable -->

```

update_user.php



```

<!-- #BeginEditable "main_content" -->
<table width="348" border="0" cellspacing="0" cellpadding="8">
<tr>
<td colspan="2">
<p><b><font color="#CCCCCC" size="4">Update a User</font></b></p>
</td>
</tr>
<tr>
<td colspan="2">
<form method="post" action="find_user.php">
<table width="400" border="0" cellpadding="5">
<tr align="left" valign="top">
<td colspan="2">
<p>Search by Student ID:<font size="2">
</font>
<input type="text" name="studentid">
</td>
</tr>

```



```
mysql_select_db($db, $link)
or die("Couldn't open $db: ".mysql_error());
```

```
$sql = mysql_query ("SELECT
student_information.Student_ID,student_information.Last_Name,student_information.First_
Name,student_information.Gender,student_information.Home_Phone,student_information.E
mail,student_information.Major,student_information.availability,
login.Username,login.Password,login.student_id FROM student_information,login WHERE
student_information.student_id LIKE '$studentid'");
$row = mysql_fetch_array($sql);
do
{
print "<p>";
print "<table width='349' border='0' cellspacing='0' cellpadding='8'>";
print "<tr>";
print "<td colspan='2'>";
print "<b><font color='#CCCCCC' size='4'>";
print "Edit User Profile";
print "</font></b>";
print "</td>";
print "</tr>";
print "<tr>";
print "<td>";
print "Student ID: ";
print "</td>";
print "<td>";
print "<input type='text' name='studentid' value='";
print $row['Student_ID'];
print ">";
print "</td>";
print "</tr>";
print "<tr>";
print "<td>";
print "Last Name: ";
print "</td>";
print "<td>";
print "<input type='text' name='lastname' value='";
print $row['Last_Name'];
print ">";
print "</td>";
print "</tr>";
print "<tr>";
print "<td>";
print "First Name: ";
print "</td>";
print "<td>";
```



```

print "<input type=\"text\" name=\"firstname\" value=\"\"";
print $row['First_Name'];
print "\">";
print "</td>";
print "</tr>";
print "<tr>";
print "<td>";
print "Gender: ";
print "</td>";
print "<td>";
print "<input type=\"text\" name=\"gender\" value=\"\"";
print $row['Gender'];
print "\">";
print "</td>";
print "</tr>";
print "<tr>";
print "<td>";
print "Major: ";
print "</td>";
print "<td>";
print "<input type=\"text\" name=\"major\" value=\"\"";
print $row['Major'];
print "\">";
print "</td>";
print "</tr>";
print "<tr>";
print "<td>";
print "Phone: ";
print "</td>";
print "<td>";
print "<input type=\"text\" name=\"phone\" value=\"\"";
print $row['Home_Phone'];
print "\">";
print "</td>";
print "</tr>";
print "<tr>";
print "<td>";
print "Email: ";
print "</td>";
print "<td>";
print "<input type=\"text\" name=\"email\" value=\"\"";
print $row['Email'];
print "\">";
print "</td>";
print "</tr>";
print "<tr>";

```

```

print "<td>";
print "Availability: ";
print "</td>";
print "<td>";
print "<input type='text' name='availability' value='\"";
print $row['availability'];
print "\>";
print "</td>";
print "</tr>";
print "<tr>";
print "<td>";
print "Username: ";
print "</td>";
print "<td>";
print "<input type='text' name='username' value='\"";
print $row['Username'];
print "\>";
print "</td>";
print "</tr>";
print "<tr>";
print "<td>";
print "Password: ";
print "</td>";
print "<td>";
print "<input type='text' name='password' value='\"";
print $row['Password'];
print "\>";
print "</td>";
print "</tr>";
print "<tr>";
print "<td>";
print "</td>";
print "<td>";
print "</table>";
//print " <input type='submit' value='Submit' /> <input type='reset' value='Reset'
/></table></form>";
} while ($row = mysql_fetch_array($sql));

?>
<table width="349" border="0" cellspacing="0" cellpadding="8">
<tr align="left" valign="top">
<td>Teachable Courses: <br>
</td>
<td>
<select name="coursename[]" multiple size=5>
<?php

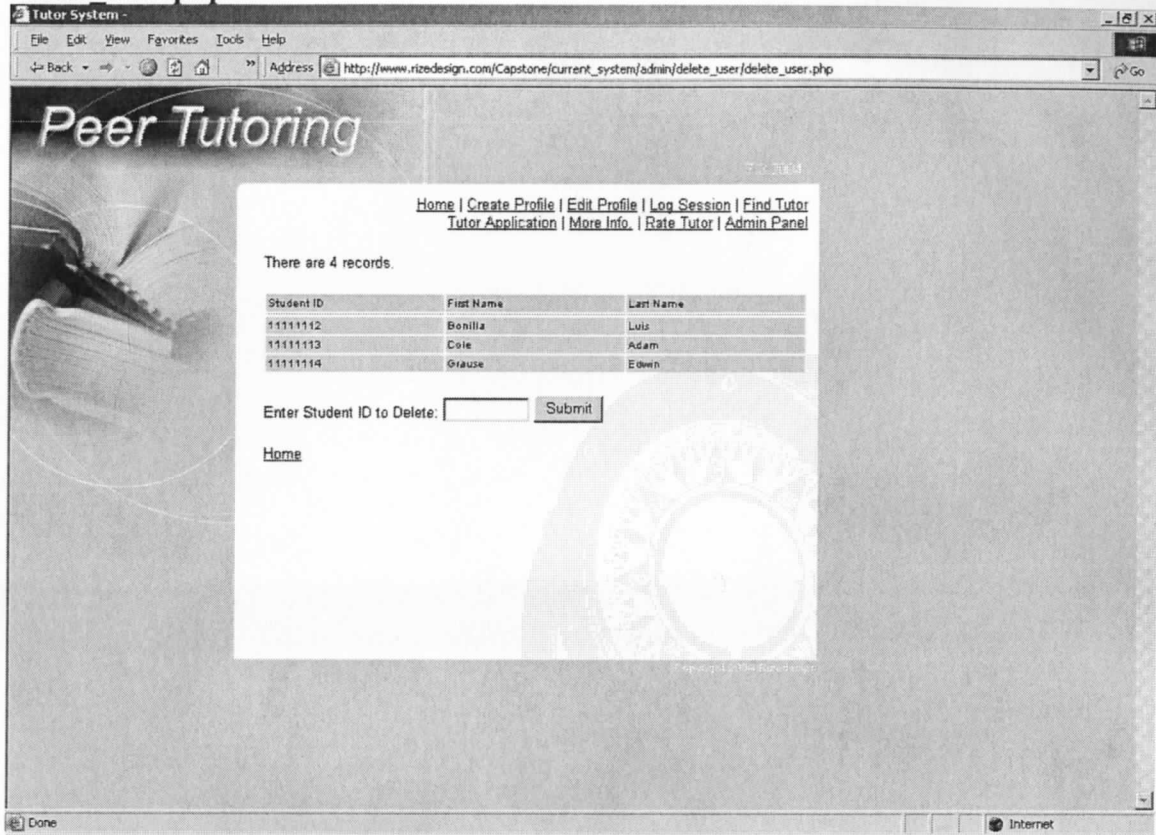
```

```

$query=("select * from course");
$result=mysql_query($query) or die ("Unable to Make the Query:" . mysql_error() );
while($row=mysql_fetch_array($result)){
echo "<OPTION VALUE=".$row['Course_ID'].>".$row['Course_Name'].</OPTION>";
}
?>
</select>
</td>
</tr>
<tr>
<td>&nbsp;</td>
<td valign="bottom">
<input type="submit" value="Submit" name="submit">
<input type="reset" value="Reset" name="reset">
<br>
</td>
</tr>
<tr>
<td>&nbsp;</td>
<td valign="bottom">
<div align="right"><a href="admin_control.php">Cancel</a></div>
</td>
</tr>
</table>
</form> <br>
<font size="1">Fields marked with an asterisk (*) are required.</font>
<?php session_destroy(); ?>
<!-- #EndEditable -->

```

delete_user.php



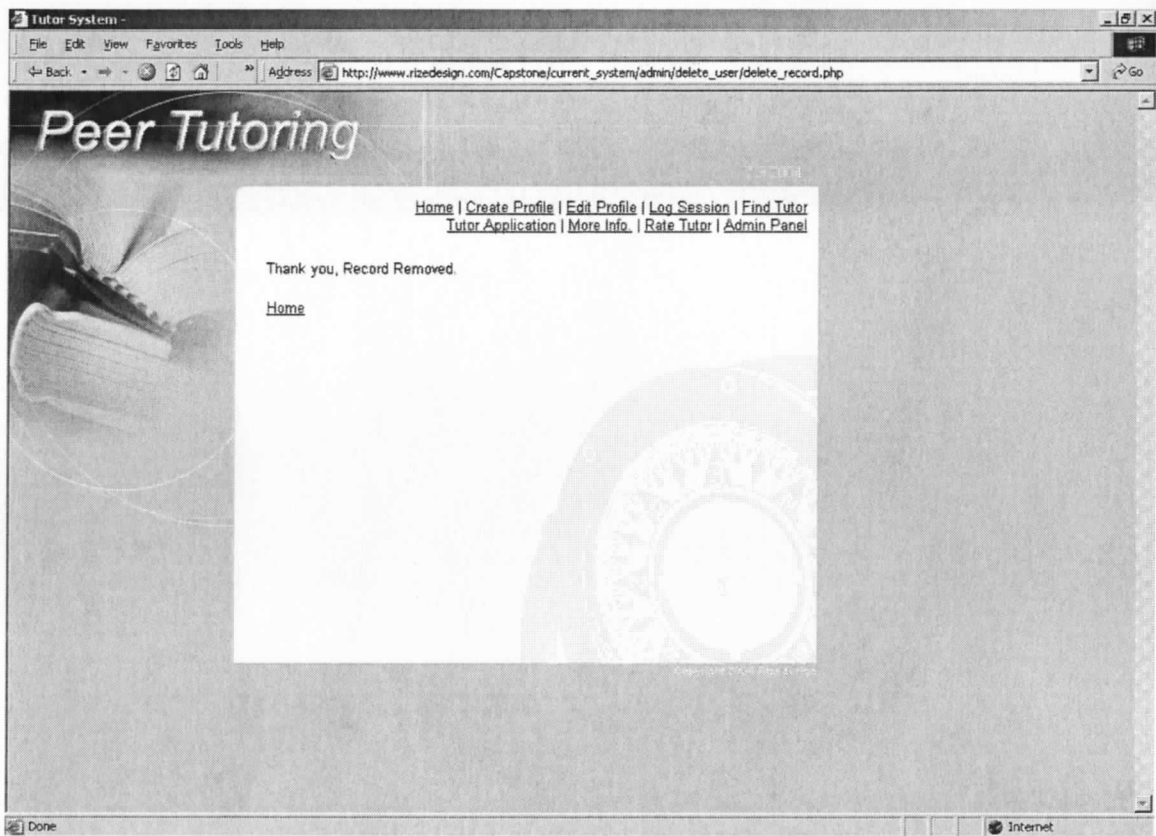
```
<!-- #BeginEditable "main_content" -->
<?
$db="wyatt1_tutorsystem";
$link = mysql_connect("localhost", "wyatt1_itt", "student");
if (! $link)
die("Couldn't connect to MySQL");
mysql_select_db($db , $link)
or die("Couldn't open $db: ".mysql_error());
$result = mysql_query( "SELECT student_id, Last_Name, First_Name FROM
student_information" )
or die("SELECT Error: ".mysql_error());
$num_rows = mysql_num_rows($result);
print "There are $num_rows records.<P>";
echo "<table width='100%'><b><tr bgcolor='\"#cccc99\"'><td><font face=arial
size=1/>Student ID</font></td><td><font face=arial size=1/>First
Name</font></td><td><font face=arial size=1/>Last Name</font></td></tr></b>";
while ($get_info = mysql_fetch_row($result)){
print "<tr bgcolor='\"#cccccc\"'><n\"";
foreach ($get_info as $field)
print "\t<td><font face=arial size=1/>$field</font></td><n\"";
print "</tr><n\"";
}
print "</table><n\"";
```

```

mysql_close($link);
?> <br>
<form method="POST" action="delete_record.php">
<p><font size="2">Enter Student ID to Delete:</font>
<input type="text" name="id" size="9">
<input type="submit" value="Submit" name="submit">
</p>
<p><a href=" ../admin_control.php">Home</a> </p>
</form>
<!-- #EndEditable -->

```

delete_record.php



```

<!-- #BeginEditable "main_content" -->
<table width="300" >
<tr>
<td> <?
$id=$HTTP_POST_VARS['id'];
$db="wyatt1_tutorsystem";
$link = mysql_connect("localhost", "wyatt1_itt", "student");

if (! $link)
die("Couldn't connect to MySQL");
mysql_select_db($db , $link)

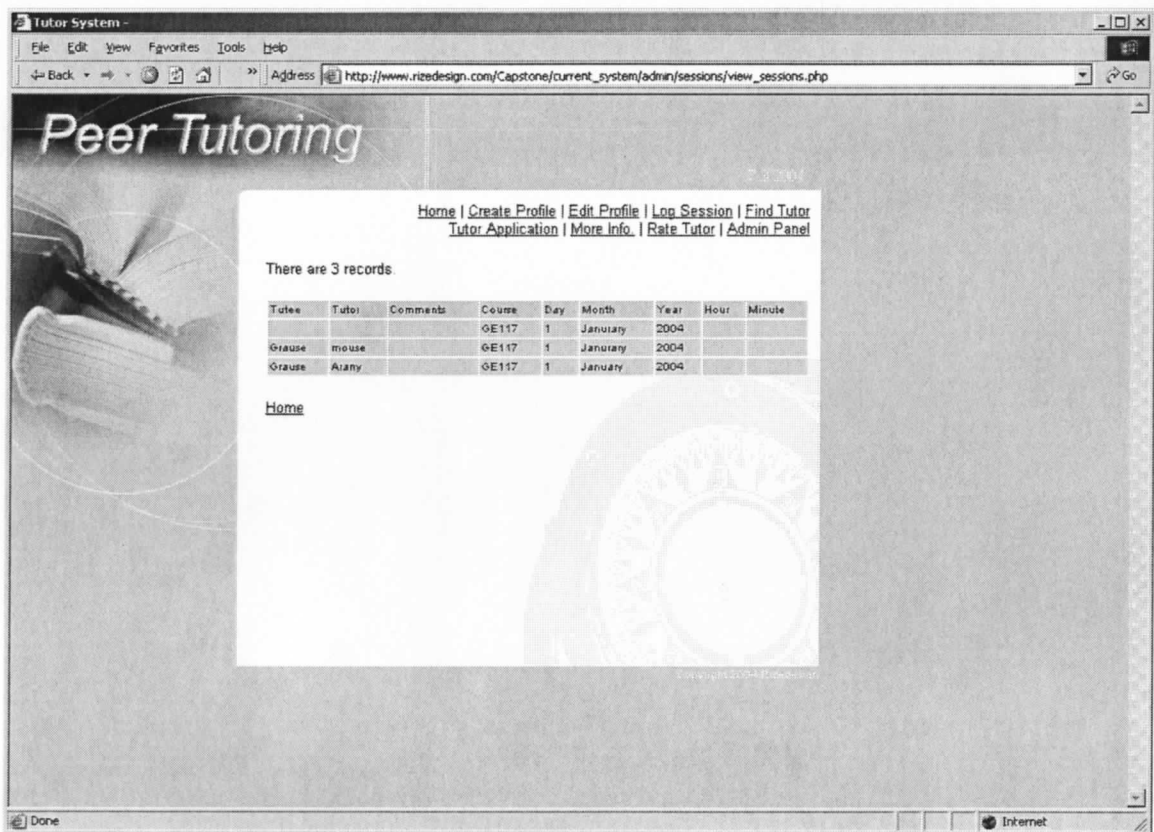
```

```

or die("Couldn't open $db: ".mysql_error());
mysql_query("DELETE FROM student_information where student_id=$id")or die("Delete
Error: ".mysql_error());
mysql_close($link);
print "Thank you, Record Removed.\n";
?>
<p><a href="../admin_control.php">Home</a> </p>
</td>
</tr>
</table>
<!-- #EndEditable -->

```

view_sessions.php



```

<!-- #BeginEditable "main_content" -->
<?php

```

```

$db="wyatt1_tutorsystem";
$link = mysql_connect("localhost", "wyatt1_itt", "student");
if (!$link)
die("Couldn't connect to MySQL");
mysql_select_db($db, $link)
or die("Couldn't open $db: ".mysql_error());

```

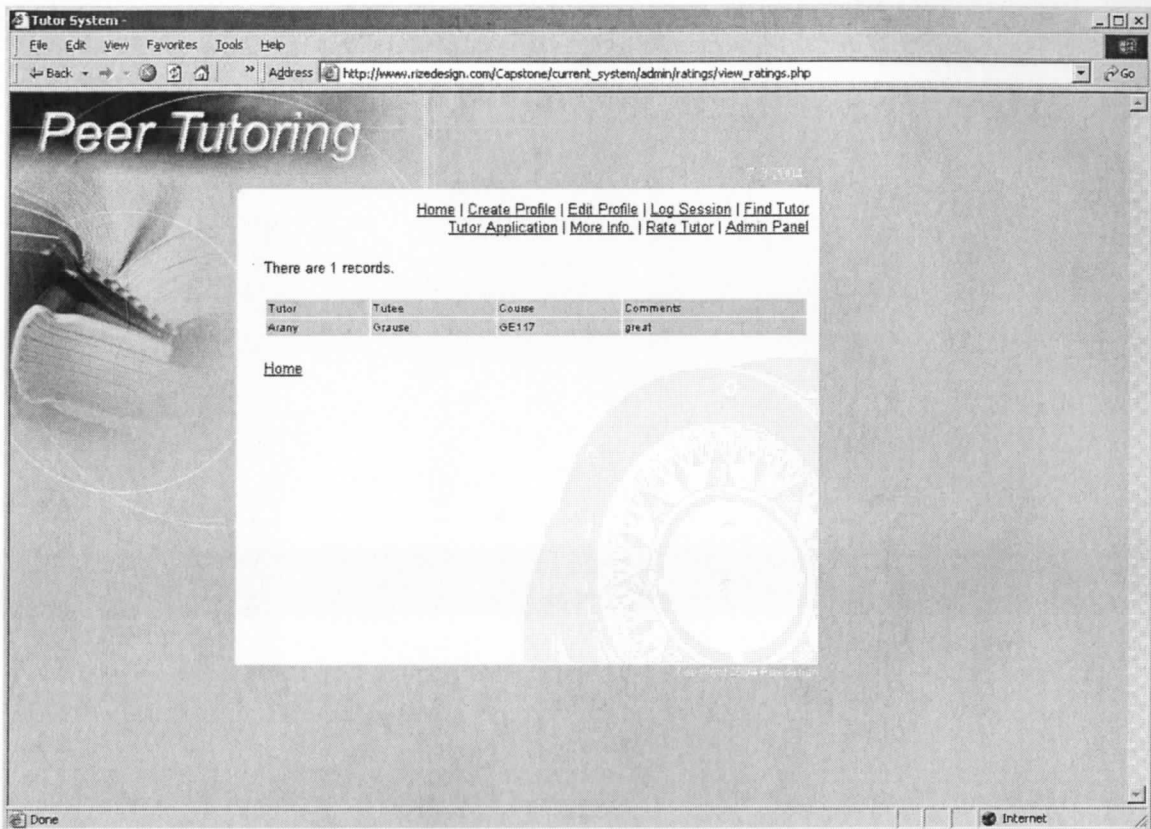
```

$result = mysql_query(
"SELECT * FROM tutor_session")

or die("SELECT Error: ".mysql_error());
$num_rows = mysql_num_rows($result);
print "There are $num_rows records.<P>";
echo "<table width='\"100%\"'><b><tr bgcolor='\"#cccc99\"'><td><font face=arial
size=1/>Tutee</font></td><td><font face=arial size=1/>Tutor</font></td><td><font
face=arial size=1/>Comments</font></td><td><font face=arial
size=1/>Course</font></td><td><font face=arial size=1/>Day</font></td><td><font
face=arial size=1/>Month</font></td><td><font face=arial
size=1/>Year</font></td><td><font face=arial size=1/>Hour</font></td><td><font
face=arial size=1/>Minute</font></td></tr></b>";
while ($get_info = mysql_fetch_row($result)){
print "<tr bgcolor='\"#cccccc\"'>\n";
foreach ($get_info as $field)
print "\t<td><font face=arial size=1/>$field</font></td>\n";
print "</tr>\n";
}
print "</table>\n";
mysql_close($link);
?>
<p><a href="\"../admin_control.php\"">Home</a></p>
<!-- #EndEditable -->

```


view_ratings.php



```
<!-- #BeginEditable "main_content" -->
<?php
```

```
$db="wyatt1_tutorsystem";
$link = mysql_connect("localhost", "wyatt1_itt", "student");
if (! $link)
die("Couldn't connect to MySQL");
mysql_select_db($db , $link)
or die("Couldn't open $db: ".mysql_error());
```

```
$result = mysql_query(
"SELECT * FROM rate_tutor")
```

```
or die("SELECT Error: ".mysql_error());
$num_rows = mysql_num_rows($result);
print "There are $num_rows records.<P>";
echo "<table width='100%'><b><tr bgcolor='\"#cccc99\"'><td><font face=arial
size=1/>Tutor</font></td><td><font face=arial size=1/>Tutee</font></td><td><font
face=arial size=1/>Course</font></td><td><font face=arial
size=1/>Comments</font></td></tr></b>";
```

```
while ($get_info = mysql_fetch_row($result)){
```



```
print "<tr bgcolor=\"#cccccc\">\n";
foreach ($get_info as $field)
print "\t<td><font face=arial size=1/>$field</font></td>\n";
print "</tr>\n";
}
print "</table>\n";
mysql_close($link);
?>
<p><a href=" ../admin_control.php">Home</a></p>
<!-- #EndEditable -->
```