Development of a Learning Management System for Ucar-Comet

Dan Riter
Regis University

Follow this and additional works at: http://epublications.regis.edu/theses
Part of the Computer Sciences Commons

Recommended Citation
Use of the materials available in the Regis University Thesis Collection ("Collection") is limited and restricted to those users who agree to comply with the following terms of use. Regis University reserves the right to deny access to the Collection to any person who violates these terms of use or who seeks to or does alter, avoid or supersede the functional conditions, restrictions and limitations of the Collection.

The site may be used only for lawful purposes. The user is solely responsible for knowing and adhering to any and all applicable laws, rules, and regulations relating or pertaining to use of the Collection.

All content in this Collection is owned by and subject to the exclusive control of Regis University and the authors of the materials. It is available only for research purposes and may not be used in violation of copyright laws or for unlawful purposes. The materials may not be downloaded in whole or in part without permission of the copyright holder or as otherwise authorized in the "fair use" standards of the U.S. copyright laws and regulations.
REGIS UNIVERSITY
SCHOOL FOR PROFESSIONAL STUDIES

MASTER OF SCIENCE
IN
COMPUTER INFORMATION TECHNOLOGY

Development of a
Learning Management System for
UCAR-COMET

PROFESSIONAL PROJECT

Dan Riter

March 6, 2006
Abstract

UCAR-COMET creates web-based training modules for professional meteorologists. Prior to the implementation of this project, UCAR-COMET did not have a Learning Management System. A Learning Management System is a web-based system that allows meteorologists to keep track of personal information that relates to each learning module. The Learning Management System also allows the meteorologists’ supervisors to monitor this information. Although there are many Learning Management Systems on the market, most are expensive, cannot be customized, and require contract agreements. UCAR-COMET would like to own a web-based system where users can create and manage their own accounts. A user account primarily keeps track of progress on each module and quiz scores. In the future, UCAR-COMET would like to add a feature that will allow the user to take notes and bookmark pages while completing a module. UCAR-COMET is a non-profit organization making cost a major factor.

A low-cost solution was achieved through the use of a MySQL database and php GUI access pages. The system is hosted on an Apache Web Server. Qualified staff built and implemented the solution. The web server is located in-house and users can gain access to modules and the Learning Management System via browser and Internet connection. The GUI is straightforward and easy to use. Support for the system is provided via email.

The Learning Management System is quickly becoming part of any web-based training program. The training budget is often the first item to get cut when money is tight. UCAR-COMET must be proactive in providing meteorologists’ supervisors with information on the effectiveness of web-based training. The effectiveness of training can
be quantified via the information tracked by the Learning Management System. Through the implementation of a low-cost Learning Management System, COMET-UCAR meets its clients’ information tracking requirements.
### Table of Contents

1.0 CHAPTER ONE: INTRODUCTION .......................................................... 9
   1.1 PROBLEM STATEMENT ................................................................. 9
   1.2 REVIEW OF EXISTING SITUATION ............................................. 9
   1.3 PROJECT GOALS .......................................................................... 11
   1.4 BARRIERS AND ISSUES ............................................................... 12
   1.5 PROJECT SCOPE .......................................................................... 12
   1.6 SUMMARY .................................................................................... 13

2.0 CHAPTER TWO: RESEARCH ................................................................. 14
   2.1 REVIEW OF AVAILABLE SOLUTIONS .......................................... 14
   2.2 HISTORY OF LEARNING MANAGEMENT SYSTEMS .................... 17
   2.3 DISCUSSION OF AVAILABLE SHAREWARE/FREeware TOOLS ....... 18
   2.4 SUMMARY .................................................................................... 20

3.0 CHAPTER THREE: PROJECT METHODOLOGY .................................... 21
   3.1 RESEARCH METHODS USED ....................................................... 21
   3.2 ANALYSIS PHASE ........................................................................ 21
       3.2.1 BUSINESS REQUIREMENTS ............................................. 22
       3.2.2 TECHNICAL REQUIREMENTS .......................................... 23
       3.2.3 SOFTWARE REQUIREMENTS ........................................... 24
       3.2.4 REPORTING REQUIREMENTS .......................................... 24
       3.2.5 TRAINING REQUIREMENTS ............................................. 25
   3.3 DESIGN PHASE ............................................................................ 26
       3.3.1 THE PROJECT PLAN ........................................................... 26
       3.3.2 THE SUPPORT PLAN .......................................................... 27
       3.3.3 THE TRAINING PLAN ......................................................... 28
   3.4 DEVELOPMENT PHASE ............................................................... 28
       3.4.1 BUILD DATABASE ............................................................... 29
       3.4.2 BUILD INTERFACE ............................................................. 29
       3.4.3 SERVER SETUP ................................................................. 30
   3.5 TESTING PHASE ........................................................................... 30
       3.5.1 TEST DATABASE ............................................................... 31
       3.5.2 TEST INTERFACE .............................................................. 31
       3.5.3 TEST SERVER ................................................................. 32
       3.5.4 DEVELOP MAINTENANCE PLAN ....................................... 32
   3.6 MAINTENANCE PHASE ............................................................... 32
       3.6.1 MANAGING UPGRADE REQUESTS ................................... 33
       3.6.2 MAINTAIN DATABASE ......................................................... 33
       3.6.3 MAINTAIN SERVER ............................................................ 33
       3.6.4 PROJECT CLOSURE ........................................................... 34
   3.7 MILESTONES ............................................................................... 34
   3.8 OUTCOMES .................................................................................. 34
1.0 Chapter One: Introduction

1.1 Problem Statement

UCAR-COMET creates web-based training modules for professional meteorologists. Prior to the implementation of this project, UCAR-COMET did not have a Learning Management System. A Learning Management System is a web-based system that allows meteorologists to keep track of personal information that relates to each learning module. This System allows the meteorologists’ supervisors to monitor this information. Although there are many Learning Management Systems on the market, most are expensive, cannot be customized, and require contract agreements. UCAR-COMET would like to own a web-based system where users can create and manage their own accounts. A user account primarily keeps track of progress on each module and quiz scores. In the future, UCAR-COMET would like to add a feature that will allow the user to take notes and bookmark pages while completing a module.

1.2 Review of Existing Situation

“In 1989 the University Corporation for Atmospheric Research (UCAR) and the National Weather Service (NWS) established the Cooperative Program for Operational Meteorology, Education and Training (COMET) to promote a better understanding of mesoscale meteorology and to maximize the benefits of new weather technologies. Today the COMET Program addresses education and training needs in the atmospheric and related sciences through online learning modules” (COMET).

An executive board and an advisory panel govern the COMET Program. The executive board is made up of the heads of the training organizations that sponsor the
program. The advisory panel is composed of representatives of the academic and operational forecasting communities that COMET serves. Examples of training modules can be found on the Meted website, UCAR-COMET’s learning module portal at http://meted.comet.ucar.edu.

Prior to the implementation of this project, UCAR-COMET did not have a Learning Management System. When a user took a module, there was no information stored about their time or of the amount of the module that was completed. Users take modules for either personal education or job requirements. Many of UCAR-COMET’s sponsors require that their employees take and pass certain learning modules. Module quizzes are emailed to management at UCAR-COMET, but no database is actively compiling these user quiz scores. Databases analyzing quiz scores are needed because the supervisors of many users would like to be able to check on their employees’ scores. In addition, these scores can be used to analyze which areas might be harder for some of their employees to grasp.

Prior to the implementation of the Learning Management System, emails sat on individual managers’ computers and nothing was ever done with them. A system was needed to compile information and scores so the managers could know that training information was in fact being learned, acquired and retained. There needed to be accountability from the user towards their completion of the modules. There was no way to run reports on these quiz scores to gain useful information. Information compiled could be used to analyze the effectiveness of training modules, how much time a user spends reading through the information, and learning vs. use of information data.
Sponsors are supporting members of the non-profit entity called UCAR-COMET. Each sponsor agrees to a fixed amount of annual funding. The amount of funding each sponsor provides determines the amount of voting power on the board. This board determines what learning modules and what special projects (such as the creation of a Learning Management System) are created in a given fiscal year. The panel decides which training modules to build based on many different trends such as new technology, popular issues, and catastrophic weather events.

1.3 Project Goals

The goals for this project were to investigate the possible Learning Management System options including purchasing an off-the-shelf system, adopting a free, open source system, or creating and owning a custom Learning Management System. The Learning Management System was needed to track progress and success on a module. The system should be easy to use for both users and administrators. Both users and administrators have varying ranges of technology aptitude. The Learning Management System needs to take into account these varying comfort levels with technology and make the analysis of data accessible to all parties. The system should be modular to allow future additions. COMET is consistently adding new technology and techniques to their modules. The Learning Management System needs to be able to be easily manipulated to include analysis of new additions and sections. All work was done in house with the current staff. Staff includes web designers, graphic artists, programmers, meteorologists, project managers, and administrators. Additional resources including staff and technology are not necessary. The project was deemed successful when the system was
operational meaning, it can collect, analyze, categorize, and send data. A six-month completion time line was established.

### 1.4 Barriers and Issues

An issue from the users’ perspective is that they are now required to provide UCAR-COMET with personal information and are subject to quiz score tracking. Personal information includes name, address, affiliation (where they work), email, and so on. Users’ scores can be analyzed, their time spent on a module is revealed, their completion of the module and the time it takes them to complete it are tracked. There is somewhat of an ethical issue at play when users are required to reveal their personal information and scores. One hurdle is getting the user to be comfortable with this process. Tracking an employee’s progress on training modules and tracking their test scores may be perceived as unethical if used as the basis of job performance reviews and advancement. If managers or administrators require all users to submit their information and it is just something that is required, this may alleviate the stress of the user. Reassurance by employers that quiz scores will not be used as job performance appraisal is advised as well.

### 1.5 Project Scope

The project was estimated to take six months to complete from analysis and design through implementation. Analysis and design occupied 1 month, development took 4 months and implementation and testing took 1 month, totaling 6 months. The exact dollar amount was the sum total of one full time programmer at 80K/yr, two 1/2-
time developers at 60K/yr, and three managers ¼-time at 80K/yr totaling 100K. This labor total represents the entire cost to UCAR-COMET, as all the software was free ware. The time lost to production was represented by a decrease in module production for the six-month period. COMET has a small labor force, but the other employees are able to efficiently continue module production.

1.6 Summary

UCAR-COMET was being asked by its sponsors to provide a Learning Management System so that module behavior could be tracked. The tracking included time spent in a module, completeness of a module, and quiz score information. All information is stored in a mySQL database. Individual module information is tracked for each user. Reports about all information stored are available for management via the web-based system interface. Reports are available by search type or custom searches can be created. The user and admin Guided User Interface (GUI) are a series of php pages (see appendix A and B for examples).
2.0 Chapter Two: Research

2.1 Review of Available Solutions

Three main solutions to tracking data were to build the Learning Management System in-house, use a free open source system, or purchase an off-the-shelf product. There are many off-the-shelf Learning Management Systems available. These off-the-shelf products range in price from free to hundreds of thousands of dollars. “Interact is an open source Learning Community Environment. It could also be called a Learning Management System, a Course Management System, or a Virtual Learning Environment” (Interact 1). Interact provides the source code for their Learning Management System as a free download. Although this is a cost attractive solution, the time involved in deciphering the source code is intensive. On the flip side of the open source offered by Interact are more expensive off the shelf Learning Management Systems. Negative aspects of these Learning Managements Systems are their initial cost and the cost to maintain effectiveness. One example of such a Learning Management System is geo-learning. According to Liz Page, “The National Weather Service recently learned the hard way that Learning Management Systems can cost a fortune. The National Weather Service pays $100K/year upkeep on the system, which includes access to SkilSoft and NetG courses. Initial cost was $250K (in 2001), and they have paid a bit more in customizations (on the order of $100K) and it's hard to estimate pain and suffering costs” (personal communication December 10, 2005).

What is the big difference between an open source system like the Interact system and a million dollar system such as geo-learning? The big difference is that there are no sales people knocking down doors to promote a free piece of software. Advertising
and marketing are expensive and are absorbed in the cost of high-priced, off the shelf software. The high-priced software product may be of a higher quality, but at a tremendously higher financial price. Another big difference is that the open source system does not come with support. Technical support for high priced software is also rolled into its cost. If a company is not technically savvy and prefers to involve itself in the business of learning about source code, then perhaps the high price is warranted. Many different quirks in free software can be frustrating for inexperienced, non-savvy employees. Technical support may be a necessity for those types of companies. If a business is willing to undergo the risks of learning and supporting an open source system then the potential cost savings may be a reward. An extensive amount of money can be saved if freeware is used. However, tech-savvy employees are needed to deal with complications that arise with open source systems. Other Learning Management System providers such as Tracorp, offer a free Learning Management System as long as the company enters into an agreement to purchase courses.

Another is that not all Learning Management Systems are created equal. Some include more functionality than others. This can be a benefit if the purchaser is not exactly sure what they want. They can play with the system and see what features might be useful. However, too much functionality can bog down the system and be burdensome. Not enough functionality leaves the system useless. The UCAR-COMET Learning Management System must contain the precise amount of functionality and be available and customizable for the future. Some of these functionality traits include tracking of page level detail, book marking pages, and web-page building tools. Many Learning Management System companies have come and gone over the past ten years as
web technologies and online learning have grown. Purchasing software leaves COMET susceptible to investing in a high priced software system only to be left without tech support when the company ceases to exist.

There are many academically known Learning Management Systems such as the one used by Regis University, WebCT. Black board, E-college, and Jones International are also widely used among the university communities. Most of these systems are built around a forum where students can collaborate. In UCAR-COMET’s case, learners are not collaborating as in a classroom environment. UCAR-COMET’s learners take modules on independently and do not interact with other learners. These academic systems are not appropriate for UCAR-COMET’s purposes.

Creating a custom Learning Management System from scratch where the source code knowledge is in the hands of the employees was attractive to UCAR-COMET. UCAR-COMET has a team already trained to create web-based training modules. Their skills include designing and developing websites, management of technical projects, instruction design, database design, and programming. Creating a Learning Management System was not above the staffs’ knowledge base. The benefits of creating a Learning Management System in-house include:

- The source code is 100% owned
- The source code is in the control of UCAR-COMET
- The system is custom designed for COMET’s needs
- Maintenance can be performed in-house
- The system resides in-house
There are no concerns about Learning Management System providers going out of business

There are no annual fees

There are no initial high cost purchases

2.2 History of Learning Management Systems

Just what is a Learning Management System? There are many different flavors of Learning Management Systems. A Learning Management System can be something as simple as a forum where students gather to communicate via text postings or as complex as a system that tracks every mouse click, every question answered, and the amount of time elapsed. The forum design is much like the Regis online environment via WebCT. As of this writing, forums have been around for more than 10 years. A forum provides a centralized place where students in specific classes can communicate, post class work, and comment about ideas or assignments. WebCT does have other features, but most Regis online classes only make use of the forum for communication. Right now, a type of forum that is popular is called a blog. A Blog (meaning weB-log) is pretty much a forum with a new name. There is nothing revolutionary about the blog, as media hype would have us believe.

As web-based training gained momentum over the last ten years, Learning Management Systems became a prevalent way to track users’ web-based training efforts. Expensive, feature-rich Learning Management Systems are driven by the corporate sector. Countless e-learning companies have come and gone leaving many abandon Learning Management Systems in their path. One example of this evolution is
intellinex.com. Intellinex is a company that evolved from several e-learning companies. Intellinex is still in business, but their original Learning Management System called LEAP is now defunct. Companies who purchased the LEAP system are left with an unsupported, dead system. The company is left at the mercy of the Learning Management System creator. This story can be found over and over again. All one has to do to see the now defunct Learning Management System trail is consult last years e-learning expo guide. The e-learning expo is an annual trade show where Learning Management System creators display their systems. From year to year the company list changes. Learning Management Systems come and go like the wind. Although the corporate sector still purchases off-the-shelf systems, free, open source Learning Management Systems are now widely available.

2.3 Discussion of available shareware/freeware tools

There are many types of shareware/freeware tools available to the consumer. Moodle is an example of a free, open source system that can be used as a Learning Management System. “It was created as a Ph.D. dissertation project, and has taken on a life of its own. Moodle is advertised as easy to install, use and maintain on Linux, Windows and Mac OS X platforms” (Berkeley). Moodle’s functionality includes forums, journals, quizzes, resources, and an area for displaying assignments. The administrative side of the system an assignment-tracking feature that time stamps a students work when uploaded to the server. The forum can be viewed as nested, flat or threaded, oldest or newest first. Moodle also has the ability to post photos with text messages. “Yahoo! Groups, Blackboard, and WebCT do not have this capability” (Berkeley). Although this
system is a good free solution, the UCAR-COMET wants to hold and maintain ownership of the system. Adopting a free system means that the UCAR-COMET technical staff must become familiar with the source code. Attempting to customize may end up being more work than building the system from scratch.

Another free Learning Management System considered by UCAR-COMET is the Jones International Learning Management System. The Jones International system is very much like the WebCT system used by Regis University. It allows different courses to be entered, forums can be established, and professors have administrative access to a grade posting area. As a real-world example of this system, the Jones International Learning Management System was successfully used by Metro College of Denver in the late 90s. During this time the Jones International system was a pay service that included support. Jones International decided to stop support of the system and it was made available as a free, open source download.

Although the Moodle Learning Management System and the Jones International Learning Management System are both free, there is still cost involved because UCAR-COMET programmers must interpret the source code in order to customize the system. Another issue is functionality of both free systems. The free systems use a forum environment that is not a desirable feature for UCAR-COMET. Ultimately, the programming staff at UCAR-COMET decided it would be more desirable to create a system from scratch where all the source code was known by the developers than to attempt to interpret a third party system. The decision to build from scratch gave UCAR-COMET the ability to include only the desired functionality. This simple functionality provided UCAR-COMET with a lightweight Learning Management System that is easily
expandable in the future. Having a staff that created the source code leaves UCAR-COMET with complete knowledge of the inter workings of the system.

2.4 Summary

There were many Learning Management Systems from which to choose. Some Learning Management Systems are open source systems that are free of charge, others cost millions of dollars. Some Learning Management Systems have many features while others have few. Not all Learning Management Systems are created equal. The features that were desirable for UCAR-COMET are minimal and include tracking time in a module, location in a module, and quiz scores. Creating a Learning Management System from scratch, in house was the desired method for UCAR-COMET because UCAR-COMET now owns the system, the system can be easily changed, and the system includes only the desired functionality.
3.0 Chapter Three: Project Methodology

3.1 Research Methods Used

The project attempted to follow the project lifecycle model including analysis, design, development, testing, implementation, and maintenance phases. Ultimately, the project used a rapid prototyping methodology meaning that the some amount, but not all of the analysis was preformed before moving ahead to the design phase, development on prototype pages were started before the design was complete, and so on. The project bounced between stages as needed. Some of the pieces like the php pages were more cyclical in nature. Small amounts of each phase were completed and then cycled through again for better flushing out of code and concepts. Overall, the project life cycle provided a good mapping of what was to be done, but reality dictated that some phases are revisited after they are complete.

3.2 Analysis Phase

During the analysis phase, project goals were refined into detailed functions and operations that the Learning Management System would exhibit. In the analysis phase, UCAR-COMET looked at the possible Learning Management System solutions. As discussed in the research section of this paper, the three main choices were to purchase an off-the-shelf system, use an open source system, or to build a custom system (Refer back to section 2.3 for details about the researched systems.). Sending a lot of money for an off-the-shelf system is not an option because UCAR-COMET is a non-profit agency with a small Learning Management System budget. Although UCAR-COMET researched using the free open source Moodle and Jones International systems, UCAR-COMET
wants the ability to own and control their system. UCAR-COMET did not want the complications of deciphering another party’s source code and did not want to have to delete a lot of superfluous functionality. Tapping into the knowledge base already employed at UCAR-COMET in order to build a custom system was the most attractive avenue. Analysis of the off-the-shelf Learning Management Systems did not take long. A major partner in UCAR-COMET, the National Weather Service, underwent an off-the-shelf Learning Management System decision that proved to be extremely costly. The National Weather Services’ work with the Geo-learning system allowed UCAR-COMET first hand knowledge and experience with a costly initial investment that continued to require substantial financial updates. This experience showed UCAR-COMET that a relationship with an off the shelf Learning Management System vendor can be expensive. Once a company is committed to a Learning Management System, it is not easy to change systems. Learning modules are linked into the system and data is stored in the database. These items are not easily transferred over to other systems due to differences in database table structure, database type, and methods for tracking web-based information.

### 3.2.1 Business Requirements

UCAR-COMET requires that the Learning Management System be simple, easy to use, and cost effective. UCAR-COMET is interested in building a custom Learning Management System in order to control its features and to have ownership of the system. The driving business factor is the request from sponsors to implement a Learning Management System. The requirements are that the system track the time spent in a
module, the completeness of a module, and test scores associated with the modules. These results are stored in the mySQL database and are available to the user and administrators via the php access pages.

3.2.2 Technical Requirements

The Learning Management System must be easy to use. To begin, the user is first required to create an account to log into the system, which will be stored in a MySQL database.

List of requirements:

- If a user’s browser crashes, the quiz score is recorded up to that point in the database
- If a user logs back in after a crash, the system ask the user if they want to resume where they left off
- A registration table stores user’s personal information
- Each quiz and module taken is associated with the user
- Courses can be created by bundling multiple modules
- Course and module completion certification is provided
- The quiz must randomize the order of the questions
- Question types must include drop-down selections and multiple choice (See appendix A)
- The system must create desired reports (See appendix B) for a variety of users
• Quiz questions can be accompanied by a graphic, Flash swf, animation, etc.

• If a quiz is taken more than once by the same individual, it is overwritten by the new answers (i.e. the user can improve their score)

3.2.3 Software Requirements

The software used to develop the UCAR-COMET Learning Management System includes MySQL open source database management software, a free download from mySQL.com, Apache open source web server software available as a free download from apache.org, and php which is “a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML (PHP.net).” Php code can be edited in a text editor program such as vi or notepad. The primary tool used to create php code is Dreamweaver, a web page editing software. SQL is used to query the database and php displays these results as output in a web-based format.

3.2.4 Reporting Requirements

Reporting of data is imperative in the success of the project. Reporting requirements include:

• The number of quizzes taken per month for each quiz

• The total head count of users registered in the system by affiliation

• Maximum, minimum, and average quiz score results

• Percentage of time an answer has been selected for each question in a quiz

• Number of people taking quizzes from the private sector
• Individual scores on quizzes
• Individual success on entire modules
• Module completion data
• Question by question analysis

3.2.5 Training Requirements

The UCAR-COMET staff members that require access to the Learning Management System are limited to a small number of people. Training on the administration side of the system was conducted in a classroom environment on site. The training time only took 1-2 hours, as the system is not complicated. Training topics include how to enter quizzes into the system and running reports on the data. The training was accomplished using a PowerPoint presentation accompanied by a brief introduction to the Learning Management System software. Many of the employees that interact with the administration pages are the same employees who built the system. These employees already have a thorough knowledge of the system. They did not need an extensive, intricate introduction to the software and its purpose.

There was no training of the user side of the system. It was assumed that users of UCAR-COMET web-based training modules already possessed the ability to interact with these few, straightforward web pages because all UCAR-COMET modules are web-based to begin with. The user experience is similar to any other web-based registration that may occur on a third-party website. However, support is available to users via an email contact at UCAR-COMET should a user be unable to complete the registration or
have other technical issues. The UCAR-COMET support person replies, in a timely fashion, to individual support requests as they are received.

3.3 Design Phase

The first step in the design phase was to gather requirements and to create a wish list based on managers’ meetings. These meetings established desired features including screen layouts, business rules, process diagrams, and pseudo code. Research showing what types of features were included in off-the-shelf Learning Management Systems was analyzed and considered. Sponsors dictated the basic requirements they desired. Those requirements consisted of a database system that would track time in a module, module completeness, and quiz scores based on each individual user.

The initial design was done in html text descriptions and hyperlinks. Ultimately, php pages were used as the GUI for both user and administrator. Style sheets controlled the layout, color, and fonts in the GUI. These style sheets were created to allow for ease of change throughout the life of the project.

UCAR-COMET’s chief database programmer completed the database design. The database programmer created SQL code, tables, and table relationships. The table layout created in mySQL can be seen in appendix D. Php code was primarily generated by Dreamweaver, a web page editing software. UCAR-COMET’s primary programmer created the php source code and tested it against the mySQL database. To see example php code see appendix C.

3.3.1 The Project Plan
The project plan is built around five phases of the project life cycle. The first phase was an analysis of the current situation and the available options. The analysis showed that UCAR-COMET was without a Learning Management System. The available options were to adopt a freeware system, purchase a system, or build a custom system. The second phase focused on the project design. Decisions about screen layout, features, and look and feel were made. The third phase was the project development. All database tables and php GUI access pages were created. The forth phase was the testing phase. All the created pages and database were tested. The database was populated with data in order to test php pages including reports. The fifth and final stage of the project was project maintenance. The maintenance included additions, corrections, and changes, as management deems necessary. The system is predicted to last at least five years but could be maintained longer depending on changes in technology and future Learning Management System trends.

3.3.2 The Support Plan

The support plan includes support of the system and support of the end user. Supporting the system is the responsibility of the database administrator at UCAR-COMET. Supporting the end user is done via email. An email contact under a technical support page is available to the end user. Email support goes to the entire IT staff, which includes five people. The IT staff responds to support questions depending on their ability to provide support for a give question. The expectation for support is an average of one email per day from users. One primary support contact responds to the users. The time commitment is expected to be less than 5% of the individual’s daily workload. Inside
users of the system such as managers receive support from the same support individual. The time commitment to support managers and staff is less than 5% of daily workload. Support totals for this one individual are estimated at less that 10% of daily total workload.

**3.3.3 The Training Plan**

The entire UCAR-COMET staff completed testing. The staff was asked to enter the system as end users and register. After registration was complete, the staff was asked to enter modules and take quizzes. The database administrator gauged the load on the server and took comments from the staff. They also looked over the data generated and the communication of that data. The comments were collected and reviewed by management. They were reviewed to make sure the system was working correctly and that it was providing accurate and desired results. Comments may be addressed or revisited depending on the collective decision of management. Adjustments were made depending on the decision of management and the IT department. IT tested the administrative side of the system by entering quizzes and linking modules.

**3.4 Development Phase**

The first part of the system to be developed was the servers. Both database and web server were configured and made operational. The next step was to set up MySQL database software on the database server. Tables, relationships, and basic SQL were created in the MySQL environment. Some minimal test data was loaded into the database to test the SQL statements. See appendix D for an example of the MySQL SQL testing
feature. Once the database environment was complete and tested, php page creation began. Building and testing the php pages against the database along the way made it easy to ensure proper functionality. After the development of the system was complete, the system was made live on the UCAR-COMET test web server. The test system can be accessed via http://deved.meted.ucar.edu. The final working system was moved to the actual web server after testing was complete http://meted.ucar.edu. The developed portion of the test site is the development server. This is a working copy that resided on the same physical server. Development, changes, and testing is always done on the testing server and then moved to the Meted side when complete.

3.4.1 Build database

The database tables and relationships were created in a mySQL environment. The mySQL environment is a php based GUI interface (see appendix C for an example). The tables and relationships are based on what was decided in the design phase. Each module has a unique identifier number that is linked to other records in the database. The module number is attached to a user after they complete a module. The quiz score is associated with a user and the module. The main three links are user, module, and quiz score.

In addition to the user information and quiz scoring that are stored in the database, each module contains quiz questions. Each learning module has a quiz with an unlimited number of questions. Each question is stored in the database. A question can be multiple choice or dropdown. See appendix C for examples of the types of questions available.

3.4.2 Build interface
The interface pages were built using php. The php code was created in Dreamweaver or hand coded in a text editor. Pages were designed based on specifications created in the design phase. The specifications include the use of a style sheet on all html and php pages in order to make future, global revision easy. Styles used provide easy to read text layout and colors consistent with the overall Meted website look and feel. See appendix D for example php code. For examples of the completed user GUI pages, see appendix A. For examples of completed administrative GUI pages, see appendix B. All pages were built with style sheets to allow for easy look and feel changes if required.

3.4.3 Server setup

The servers are UNIX servers. One server is the web server where Apache web server software and all the GUI php interface pages are located. The second UNIX server is the mySQL database server. The system consists of two physical server environments. The web server is open to the web and therefore more venerable to hackers. The mySQL database resides inside the firewall and is therefore safer. Only the database administrator has access to the mySQL table structure. The systems are backed up via UCAR-COMET’s existing tape back up system, which is responsible for backing up all UCAR-COMET learning module work.

3.5 Testing Phase

Testing brought all the pieces together into a special testing environment. A click through of all user and administrator GUI pages was conducted. The click through ensures that there are no broken links and that links take the user to the proper location.
Test data was loaded into the database so that test reports could be run. Dreamweaver debugging feature was used to test php code for bugs. The UCAR-COMET staff was involved in the testing phase. Every employee was asked to register and take quizzes in the system. This load test attempted to emulate the live user experience. After the in house testing was complete, reports were run to see the test totals.

3.5.1 Test database

Testing the database involves three different levels of testing. First, loading test data confirms the functionality of the database table structure. Second, reports are run on the test data to prove the expected results of the data returned. Third, in addition to report testing, load testing is performed. The UCAR-COMET staff does the load testing. The staff accesses the user GUI pages all at the same time in order to test performance of the database. Testing make sure that data is compiled and presented correctly, the server can handle the traffic, and determines whether there are any other system bugs to report.

3.5.2 Test interface

The IT staff does interface testing to ensure that all fields, links, and buttons in the php pages are functional first. This is called the click through. The UCAR-COMET staff performs a second test and uses similar techniques in a more random fashion. Any issue is reported to and fixed by IT. Comments on the interface layout are also taken. Management discusses these comments and any changes are decided upon by their team. The IT staff tests administrative pages. Multiple IT staff personnel access all php pages. The IT staff enters quizzes into the database via the administrative php pages (see
appendix B for examples). All errors are reported to the database administrator and the errors are corrected. When an error is corrected, the database administrator emails the IT staff and asks them to test the affected section again. The process continues in a cyclical fashion. Once the IT staff deems the pages usable, testing is complete.

3.5.3 Test server

Testing the server was accomplished through the interface and database testing. When the test of the interface and database were complete and successful, the server was deemed functional. Web server stats were also monitored using Apache’s built in website statistics software.

3.5.4 Develop maintenance plan

The UCAR-COMET database administrator conducts long-term maintenance, which involves periodic software upgrades as the manufactures of the software make updates available. MySQL updates are completed, as newer versions are available. Occasional security updates of the Apache web server and php code are also completed. Comments and suggestions are continuously reviewed, considered, and adjusted as needed.

3.6 Maintenance Phase

As new releases of MySQL become available, the on staff database administrator upgrades the database. The business plan calls for the system to be operational for at least five years. Hardware maintenance on the UNIX server may be required during this time.
These can include monitoring disk space or simply rebooting. Change orders are expected as ongoing manager meetings and user feedback is gathered. Correction of code may occur as bugs are detected. The database administrator at UCAR-COMET conducts primary, long-term maintenance.

### 3.6.1 Managing upgrade requests

Upgrade requests are given to the database administer by management. All upgrade requests are filtered through management. Management is responsible for approving any upgrade request. If an upgrade is approved, the IT staff determines the appropriate person on staff to conduct the update depending on their skills.

### 3.6.2 Maintain database

The database administrator performs database maintenance. Maintenance includes updates to the mySQL software as newer versions become available, purging old data, and conducting database backup to ensure database optimization.

### 3.6.3 Maintain server

UNIX server maintenance is almost non-existent. The UNIX server up time is so reliable it is virtually a maintenance-free machine as compared to other server environments such as Windows, which have a tendency to need constant rebooting. The UNIX server environment is expected to support beyond the life of the project. The UCAR-COMET network administrator performs maintenance, should maintenance be
necessary. The UCAR-COMET IT staff is already familiar with and prefers the UNIX environment.

### 3.6.4 Project closure

Completion of the Learning Management System marks official project closure although system maintenance and upgrades continue throughout the life of the system. Testing was the last phase before closure was accomplished.

### 3.7 Milestones

Project milestones include:

- Completion of the analysis phase
- Completion of the design phase
- Completion of the development phase
- Completion of the testing phase
- Project launch which marks the beginning of the maintenance phase

### 3.8 Outcomes

Project outcomes include:

- Functional GUI php user pages
- Functional GUI php administrator pages
- Functional mySQL database
- Sponsors approval of the Learning Management System
- Management approval of the Learning Management System
4.0 Chapter Four: Project History

4.1 How the project began

As Learning Management Systems became the norm for web-based training, UCAR-COMET’s sponsors requested the implementation of a Learning Management System in order to better track its employees’ learning module progress and comprehension. Since the National Weather Servers had a negative experience with an off-the-shelf Learning Management System, it was deemed that building a custom system would be the best solution for UCAR-COMET. Support for building the custom Learning Management System came from UCAR-COMET’s sponsors and management.

4.2 How the project was managed

Three top-level managers at UCAR-COMET managed the project. These managers directed the analysis, design, development, testing, implementation, and maintenance of the Learning Management System. They coordinated communication and cooperation between different employees and departments within UCAR-COMET to complete the Learning Management System.

Overall, the management team was effective. The Learning Management System was completed on time and within expected budget. The technical nature of UCAR-COMET’s business allowed the development staff to work autonomously for much of the project life cycle.

Now that the project is complete, the IT staff and UCAR-COMET management communications have reduced to an as needed basis.
4.3 Was the project considered a success?

The project was a success. Currently there are more than 2000 users registered in the UCAR-COMET Learning Management System. Those 2000 users have taken more than 9000 quizzes to date. There are more than 100 modules and quizzes available in the Learning Management System. (http://meted.ucar.edu).

The php pages (GUI) are simple and easy to use. The mySQL database has handled the user load. The system up time is acceptable and rarely requires reboot.

Support occasionally receives a complaint about the storage of personal information and the tracking of quiz scores. However, the requirement to provide personal information is only necessary if the employer is requiring that the quiz score and progress be tracked. Support is directed to inform the person that they should take the issue to their supervisor. Module and quiz access is available without registration as long as the person does not need information tracked for job requirements.

4.4 What changes occurred to the plan?

Report requirements changed throughout the project life cycle. Ultimately, a “create your own report” feature was added to try to accommodate any reporting request (See appendix B). The GUI screen layout changed continually and is still being refined today. The screen layouts have become part of the ongoing maintenance of the system. Since there were many testers at UCAR-COMET looking at the GUI pages, many comments were collected about the overall appearance. The appearance ended up being altered numerous times during the project life cycle. However, the look and feel, the
appearance of the GUI pages, is in the eye of the beholder and is likely to continue to evolve.

4.5 How did the project end?

Although the project went live and was considered a success, it continues to evolve to this day. Testing was effective, but bugs are occasionally discovered and resolved. UCAR-COMET sponsors are happy with the Learning Management System and continue to add modules as required job training is enhanced. As more people require the UCAR-COMET training, the user base will grow. The current user base of more than 2000 has exceeded the expectations of UCAR-COMET sponsors and managers.

4.6 What went right and what went wrong?

The completion of the physical system was a success. All the php coded GUI pages are acceptable and functional (see appendix A & B). The MySQL database is acceptable and functional (see appendix D). The functioning database and php pages are examples of things that went right. Managing change is hard for any project and UCAR-COMET’s managers are responsible for managing all the change to the Learning Management System. An example of something that went wrong is the change management plan. Unfortunately, the IT department must respond to all change requests that management delivers. Better planning of requirements may have lessened the amount of change requests post project completion.

The Learning Management system was completed on time and on budget. Although the time line was not a diving force, the system was operational within the
expected time frame. As stated above, changes to the system are continually being made. As feedback from users comes to management’s attention, change requests are discussed and then decisions on changes are given to the programming staff. Most changes are small and do not require much time to implement. Examples of small changes include style sheet changes that affect the look and feel of the user and administrative pages.

4.7 Project Summary

The UCAR-COMET Learning Management System is successful because it serves the intended purpose, which is to track a user’s progress through a module, their completeness of the module, and their quiz score related to the module. The timeline was kept and regularly scheduled work was not interrupted any more than anticipated. Currently there are more than 2000 users registered in the UCAR-COMET Learning Management System. Those 2000 users have taken more than 9000 quizzes to date. There are more than 100 modules and quizzes available in the Learning Management System. (http://meted.ucar.edu). UCAR-COMET’s sponsors are satisfied with the Learning Management System and are able to now gauge an employee’s progress through the module tracking features.

UCAR-COMET’s programming staff continues to maintain and implement changes to the system. One primary programmer is responsible for handling the maintenance and change whether that means working independently or requesting other IT staff’s help.
5.0 Chapter Five: Lessons Learned

5.1 What was learned from the project experience?

After completion of a working, implemented system, the project was deemed a success. It was quickly learned that not all functionality was designed into the system, however. Additional reports were requested after competition. Specifically, the report that shows how many private sector users are accessing modules was added late in the project life cycle. In addition, the ability to create a custom report was added later in the project life cycle. These additional reports were simple enough to generate, as they only required SQL to be performed on the database and php access page to display results.

The look and feel continues to change as users and managers offer feedback on the current system. UCAR-COMET has a tendency to react too quickly. If one user makes a comment about the legibility of a page, the page is immediately changed to attempt to make that one user happy. This may not be a sustainable maintenance plan. According to UCAR-COMET policy, a change request from a user must occur more than one time in order for it to be up for consideration. Then, the management team and IT must agree that the change is warranted before a change is implemented.

Occasional database errors cause users quiz scores to be dropped. This problem cannot be fully confirmed and is based on the occasional support email from a user claiming to have taken a quiz were no score is found. Since the nature of technology is not foolproof, adjustments are made to quiz scores on the honor system. Perhaps with an enterprise database system such as Oracle, dropped data would not be expected. However, with the use of the free MySQL database, occasional errors are tolerated. The
database administrator has come to the conclusion that restarting the database on a monthly basis seems to help prevent lost data. This issue is being looked into further.

As newer versions of software have come available they have been upgraded on the system. One example of were this became problematic is with the latest version of php. When the new software was installed on the system all php pages were temporarily unavailable. Although a work around was quickly established, a proper understanding of the effects in the newer version were to blame.

UCAR-COMET learned that purchasing a Learning Management System was expensive, using an open source system was risky, and building a custom system was the desired path. The custom system solution allowed UCAR-COMET to construct a simple system that only had the desired functionality. Working with a small on-staff team made changes to the system easier than dealing with out sourcing. Because the project was small and the team was small, impromptu meetings could be held to discuss the direction of the Learning Management System. It was learned that this small project and team were a very manageable task.

However, even though the project is complete and operational, change requests continue to be generated. Just as with any project, it can continue on forever if the powers allow it to be so. Although the small size of the project and team are a benefit in some regards, it is a hindrance when it comes to change requests. The small team environment makes it easy for managers to request changes on the fly. Since it is known that the system is relatively simple, changes are also perceived as simple. Learning how to live with this process for the life of the project is between the managers and the support staff who must implement the change.
5.2 What should have been done differently?

There were many lessons learned while planning and initiating this project. Most importantly, more requirements and parameters should have been created up front. Greater detail in the user and administrative page layout might limit the numerous changes that are occurring based on one or two users’ opinions. In addition, the more stringently set the layout and parameters were, the fewer and less necessary changes became warranted. Too often systems are built and only then do managers realize what they really wanted is something very different. If more time were spent during the planning phase on important details, managers would have been required to make binding decisions at that time so fewer changes were needed after development.

The UCAR-COMET Learning Management System was built using flexible technology. Making changes to php layout or adding tables to the mySQL database are not usually difficult tasks. However, as more and more parties become involved in the change request process, ideas frequently clash and the work becomes more tedious and time is wasted.

System requirements should have been better defined. Features were added to the project scope towards the end of the project. One example of a significant change that occurred late in the project is the addition of course level tracking. Originally, modules were the top-level for tracking. It was decided late in the project that modules should be able to be lumped together to form a course. Although this functionality was added to the user interface pages, (see appendix A for an example) the reports are unable to search on a course level and if a user is at the module level, there is no indicator to let them know
that the module is also part of a course. The integration of the course level element came late in the project and usability of the feature suffers because of this fact.

Another feature that is currently being discussed has to do with the end of quiz scoring display. Right now, the user’s quiz score is displayed for them and they can see which question they got correct or incorrect. The feature that would make this function better is the addition of linking back to module content that relates to each question. Currently there is no way to link into the content from a quiz question. If a user could link to the area of content, they could review the material and discover why they missed a question. This adds instructional value that was overlooked in the initial build. The addition of the feature is currently under review.

5.3 Did the project meet expectations?

The project did meet the layout requirements by sponsors and management. The system is up and running and has thus far required little maintenance. Reports can be generated to meet the needs of management. Users have been able to access and register on the system without too much trouble. Occasionally, users’ might forget their passwords, so emails are sent to request help. Users also have trouble with quizzes. Because the quizzes are web-based, technical problems can occur such as user browser’s crash.

These project goals were accomplished:

- Created, owns and maintains the Learning Management System
- Track progress and success on a module
- Easy to use for both users and administrators
- Modular to allow future additions
- Work was done in house with the current staff
- Can collect, analyze, categorize, and send data.
- A six-month completion time line met.

5.4 What is the next stage of the project?

The next step of the project is to continue evaluating the system in order to determine the next appropriate level of functionality. UCAR-COMET management will review comments from users as they come in to determine where common themes indicate a need for change.

5.5 Summary

UCAR-COMET’s sponsors requested the addition of a Learning Management System to track users learning module activity. UCAR-COMET responded by providing a fully functional Learning Management System that met all requirements. All Business Requirements, Technical Requirements, Software Requirements, Reporting Requirements, and Training Requirements were met. The existing UCAR-COMET technical staff built the Learning Management System. UCAR-COMET learned from the mistakes of the National Weather Service and avoided the high cost of an off-the-shelf Learning Management System. UCAR-COMET now wholly owns and maintains the custom Learning Management System. Sponsors, managers, and UCAR-COMET staff are all pleased with the Learning Management System.
REFERENCES


Appendix A: The User Experience
Access Point

The user gains access into the UCAR-COMET Learning Management System via this simple GUI front page. The user must create an account if one has not already been created.
Registration

Standard information is collected from the user and stored in the mySQL database.
Welcome

Once the registration is complete, the user is welcome into the system.
User Options

When a user logs into the system they have several options shown below.
User Records

A user can view a record of all quiz information stored. In this example there is one record stored.
Module and Quiz list

The user can now access any of the learning modules and quizzes available at UCAR-COMET.
Log in Message

This list of modules and quizzes is available even if the user does not create an account. The user is informed that their information will not be tracked.
Sample Question

Below is an example multiple-choice question from one of the UCAR-COMET quizzes. The user has the option to make a selection and save or skip the question. If the question is skipped, the user will be reminded to answer the question before the quiz is completed. This message occurs on the final question page.
**Quiz Results**

Once a quiz is complete the user is give a score, a list of questions, and the correct/incorrect scoring.

![Image of quiz results](image-url)
Appendix B: The Administrator Experience
Create or Edit Quizzes

This admin access screen allows an admin to create a new quiz or edit an existing quiz.

<table>
<thead>
<tr>
<th>Quiz Name</th>
<th>Deved Active</th>
<th>MetEd Active</th>
<th>Creator</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLAC Lesson 1a - Fog and Stratus Forecast Approaches</td>
<td>yes</td>
<td>yes</td>
<td>Dan Riter</td>
</tr>
<tr>
<td>DLAC Lesson 1b - Radiation Fog</td>
<td>yes</td>
<td>yes</td>
<td>Vickie Johnson</td>
</tr>
<tr>
<td>DLAC Lesson 2a - Synoptic Weather Considerations: Forecasting Fog and Low Stratus</td>
<td>yes</td>
<td>yes</td>
<td>Vickie Johnson</td>
</tr>
<tr>
<td>DLAC Lesson 2b - Local Influences on Fog and Low Stratus</td>
<td>yes</td>
<td>yes</td>
<td>Vickie Johnson</td>
</tr>
<tr>
<td>DLAC Lesson 2c - Assessing Climatology in Fog/Stratus Forecasting</td>
<td>yes</td>
<td>yes</td>
<td>Vickie Johnson</td>
</tr>
<tr>
<td>DLAC Lesson 2d - Applying Diagnostic and Forecast Tools to Forecasting Fog and Stratus</td>
<td>yes</td>
<td>yes</td>
<td>Vickie Johnson</td>
</tr>
<tr>
<td>DLAC Lesson 2e - Case Study: A New England Fog Event</td>
<td>yes</td>
<td>yes</td>
<td>Seth</td>
</tr>
<tr>
<td>DLAC Lesson 3a - Customer Impacts: Forecasting Fog and Low Stratus</td>
<td>yes</td>
<td>yes</td>
<td>Vickie Johnson</td>
</tr>
</tbody>
</table>
New Quiz

The first step in creating a new quiz is to enter some basic information about the quiz. This page has been customized to represent the meteorological learning modules UCAR-COMET provides.
**Question Type**

There are several types of question that can be created. These include: Multiple-choice with a single answer, multiple-choice with multiple answers, drop-down box at the beginning or end of a sentence, and drop-down boxes throughout a paragraph.
Delete a Question

After a quiz is in the database, questions can be deleted by selecting the desired check box and clicking the submit button.
Question Entry

Questions are entered using simple text-entry fields with prompts.

What conditions are favorable for the development and maintenance of both radiation and advection fog events?

User instructions for answering the question - tags will be formatted as html in quiz
Choose <b>all</b> that apply.

Should these possible answers be presented in random order in the quiz?

Enter possible answer #1.
Subsidence</p>

Enter possible answer #2.
Low-level moisture</p>

Enter possible answer #3.
Strong surface winds</p>

Enter possible answer #4.
Capping inversion
User instructions for answering the question - tags will be formatted as html in quiz.
Choose <code>checkbox</code>s that apply.

Should these possible answers be presented in random order in the quiz?

Enter possible answer #1
Suface aloft

Enter possible answer #2
Low-level moisture

Enter possible answer #3
Strong surface winds

Enter possible answer #4
Capping inversion

Enter possible answer #5
Strong convective mixing within the boundary layer

Enter the correct answer (e.g. 1, 2, 4)
1, 2, 4

4 points for this question - usually the same as possible answers
5
Reports

Many different reports can be generated with admin privileges.

Many reports are canned or custom reports can be created.
Canned Reports

Here is a list of the current canned reports.

MetEd Canned Registration and Quiz Reports

- **Number of quizzes taken report**
- **Count of registered students by affiliation report**
- **Minimum, maximum and average scores report**
- **Percentage of times an answer has been selected report**
- **Count of Private Sector students by Quiz Report**
**Number of Quizzes**

Once a report type is selected, the admin can choose to search by all or specific leaning module.
Number of Quizzes Taken Report

This represents a report on the number of quizzes taken each month for the DLAC Lesson 1a.

Number of Quizzes Recorded Report

DLAC Lesson 1a - Fog and Stratus Forecast Approaches

<table>
<thead>
<tr>
<th>Month</th>
<th>Quizzes</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Date</td>
<td>452</td>
</tr>
<tr>
<td>July 2004</td>
<td>43</td>
</tr>
<tr>
<td>August 2004</td>
<td>75</td>
</tr>
<tr>
<td>September 2004</td>
<td>47</td>
</tr>
<tr>
<td>October 2004</td>
<td>39</td>
</tr>
<tr>
<td>November 2004</td>
<td>49</td>
</tr>
<tr>
<td>December 2004</td>
<td>39</td>
</tr>
<tr>
<td>January 2005</td>
<td>12</td>
</tr>
<tr>
<td>February 2005</td>
<td>18</td>
</tr>
<tr>
<td>March 2005</td>
<td>30</td>
</tr>
<tr>
<td>April 2005</td>
<td>33</td>
</tr>
<tr>
<td>May 2005</td>
<td>16</td>
</tr>
<tr>
<td>June 2005</td>
<td>20</td>
</tr>
<tr>
<td>July 2005</td>
<td>22</td>
</tr>
<tr>
<td>August 2005</td>
<td>18</td>
</tr>
<tr>
<td>September 2005</td>
<td>12</td>
</tr>
<tr>
<td>October 2005</td>
<td>17</td>
</tr>
<tr>
<td>November 2005</td>
<td>14</td>
</tr>
<tr>
<td>Total</td>
<td>964</td>
</tr>
</tbody>
</table>
Count of Registered Students by Affiliation Report

This example report shows the total count of registered users based on affiliation type.

<table>
<thead>
<tr>
<th>Affiliation</th>
<th>Registered Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>not indicated</td>
<td>1</td>
</tr>
<tr>
<td>not indicated</td>
<td>277</td>
</tr>
<tr>
<td>air force</td>
<td>135</td>
</tr>
<tr>
<td>canadian government</td>
<td>65</td>
</tr>
<tr>
<td>fake</td>
<td>2</td>
</tr>
<tr>
<td>instructor</td>
<td>125</td>
</tr>
<tr>
<td>international faculty or student</td>
<td>51</td>
</tr>
<tr>
<td>international government employee</td>
<td>41</td>
</tr>
<tr>
<td>met service of canada</td>
<td>37</td>
</tr>
<tr>
<td>navy/marine corps</td>
<td>239</td>
</tr>
<tr>
<td>nona</td>
<td>673</td>
</tr>
<tr>
<td>other</td>
<td>254</td>
</tr>
<tr>
<td>other us government</td>
<td>34</td>
</tr>
<tr>
<td>private sector</td>
<td>149</td>
</tr>
<tr>
<td>us faculty or student</td>
<td>44</td>
</tr>
<tr>
<td><strong>Total Registered Users</strong></td>
<td><strong>2127</strong></td>
</tr>
</tbody>
</table>
Minimum, Maximum and Average Scores Report

This report shows the number of quizzes recorded, the max score, the min score, and the average score.

<table>
<thead>
<tr>
<th>Quiz Name</th>
<th>Recorded</th>
<th>Max%</th>
<th>Min%</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Convective Storm Matrix</td>
<td>14</td>
<td>96</td>
<td>16</td>
<td>73</td>
</tr>
<tr>
<td>A Social Science Perspective of Flood Events</td>
<td>7</td>
<td>92</td>
<td>56</td>
<td>76</td>
</tr>
<tr>
<td>An MCS Matrix</td>
<td>4</td>
<td>81</td>
<td>43</td>
<td>70</td>
</tr>
<tr>
<td>Blowing Snow - Baker Lake, Nunavut, 4-10 February 2003</td>
<td>15</td>
<td>100</td>
<td>45</td>
<td>75</td>
</tr>
<tr>
<td>CAMEO/HYSPLIT Dispersion Models</td>
<td>4</td>
<td>100</td>
<td>78</td>
<td>89</td>
</tr>
<tr>
<td>Case Studies Using the GOES Imager</td>
<td>4</td>
<td>84</td>
<td>33</td>
<td>60</td>
</tr>
<tr>
<td>Coastally Trapped Wind Reversals</td>
<td>12</td>
<td>96</td>
<td>18</td>
<td>72</td>
</tr>
<tr>
<td>Cold Air Damming</td>
<td>49</td>
<td>96</td>
<td>29</td>
<td>79</td>
</tr>
<tr>
<td>Creating a Local Climate Product Using Composite Analysis</td>
<td>15</td>
<td>100</td>
<td>36</td>
<td>79</td>
</tr>
<tr>
<td>Definition of the Mesoscale</td>
<td>12</td>
<td>94</td>
<td>50</td>
<td>81</td>
</tr>
<tr>
<td>Dispersion Basics</td>
<td>12</td>
<td>100</td>
<td>50</td>
<td>81</td>
</tr>
<tr>
<td>DLAC Lesson 1a - Fog and Stratus Forecast Approaches</td>
<td>486</td>
<td>100</td>
<td>35</td>
<td>82</td>
</tr>
<tr>
<td>DLAC Lesson 1b - Radiation Fog</td>
<td>764</td>
<td>100</td>
<td>30</td>
<td>89</td>
</tr>
<tr>
<td>DLAC Lesson 2a - Synoptic Weather Considerations: Forecasting Fog and Low Stratus</td>
<td>720</td>
<td>100</td>
<td>29</td>
<td>88</td>
</tr>
<tr>
<td>DLAC Lesson 2b - Local Influences on Fog and Low Stratus</td>
<td>695</td>
<td>100</td>
<td>33</td>
<td>88</td>
</tr>
<tr>
<td>DLAC Lesson 2c - Assessing Climatology in Fog/Stratus Forecasting</td>
<td>647</td>
<td>100</td>
<td>50</td>
<td>86</td>
</tr>
<tr>
<td>DLAC Lesson 2d - Applying Diagnostic and Forecast Tools to Forecasting Fog and Stratus</td>
<td>643</td>
<td>100</td>
<td>64</td>
<td>90</td>
</tr>
<tr>
<td>DLAC Lesson 2e - Case Study: A New England Fog Event</td>
<td>272</td>
<td>100</td>
<td>75</td>
<td>86</td>
</tr>
</tbody>
</table>
### Percentage of Times an Answer Has Been Selected Report

The report shows the percentage of time a user selected each answer. The correct answers are shown in bold. This report allows UCAR-COMET to see which questions everyone is getting correct and which are being missed.

---

#### A Social Science Perspective of Flood Events

Correct answers appear in bold

<table>
<thead>
<tr>
<th>Number of completed quizzes considered for the report: 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Prior to the Big Thompson flood event, most government policy on flooding was based on _____ flooding and not _____ flooding events.</td>
</tr>
<tr>
<td>0% - main stem river</td>
</tr>
<tr>
<td>0% - flash</td>
</tr>
<tr>
<td>71% - slow-rise</td>
</tr>
<tr>
<td>14% - flash</td>
</tr>
<tr>
<td>14% - small stream</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2. Social science studies on flash flooding have found which of the following about warnings?</th>
</tr>
</thead>
<tbody>
<tr>
<td>0% - People generally respond immediately</td>
</tr>
<tr>
<td>100% - People have to assess their own risk before responding.</td>
</tr>
<tr>
<td>71% - Warnings are not a stimulus response action.</td>
</tr>
<tr>
<td>57% - Warnings do not increase the likelihood of a response.</td>
</tr>
<tr>
<td>25% - Warnings are typically viewed by the public as a &quot;cry wolf&quot; or overstated product.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3. For a warning to provoke a desired and positive response by the public, the product should take into account which of the following?</th>
</tr>
</thead>
<tbody>
<tr>
<td>86% - Social considerations</td>
</tr>
<tr>
<td>57% - Scientific components</td>
</tr>
<tr>
<td>86% - Detection and response</td>
</tr>
</tbody>
</table>
Count of Private Sector Students by Quiz Report

This report shows the number of times a quiz was taken by someone in the private sector.
Create Your Own Report

This is the screen where a custom report can be created. All search criteria are available and can be selected as desired.
Example Results for Search by Division NOAA

From the screen above, the report shows all data in the system with a noaa.gov email address.

<table>
<thead>
<tr>
<th>Email</th>
<th>Name</th>
<th>Course Title</th>
<th>Grade</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:anne.fish@noaa.gov">anne.fish@noaa.gov</a></td>
<td>Anne Fish</td>
<td>Lesson 1a - Fog and Status Forecast Approaches</td>
<td>57</td>
<td>11/17/05</td>
</tr>
<tr>
<td><a href="mailto:al.hong@noaa.gov">al.hong@noaa.gov</a></td>
<td>Alon Hong</td>
<td>Lesson 1a - Fog and Status Forecast Approaches</td>
<td>Incomplete</td>
<td>no 4/25/05</td>
</tr>
<tr>
<td><a href="mailto:alan.cope@noaa.gov">alan.cope@noaa.gov</a></td>
<td>Not given</td>
<td>Pop Currents: Nearshore Fundamentals</td>
<td>86</td>
<td>1/3/05</td>
</tr>
<tr>
<td><a href="mailto:alan.rezek@noaa.gov">alan.rezek@noaa.gov</a></td>
<td>Alan Rezek</td>
<td>Lesson 1a - Fog and Status Forecast Approaches</td>
<td>85</td>
<td>6/5/05</td>
</tr>
<tr>
<td><a href="mailto:alan.rezek@noaa.gov">alan.rezek@noaa.gov</a></td>
<td>Alan Rezek</td>
<td>Lesson 2a - Synoptic Weather Considerations: Forecasting Fog and Low Stratus</td>
<td>88</td>
<td>6/13/05</td>
</tr>
<tr>
<td><a href="mailto:alex.calderon@noaa.gov">alex.calderon@noaa.gov</a></td>
<td>Alex Calderon</td>
<td>Lesson 1a - Fog and Status Forecast Approaches</td>
<td>999</td>
<td>6/2/06</td>
</tr>
<tr>
<td><a href="mailto:alex.calderon@noaa.gov">alex.calderon@noaa.gov</a></td>
<td>Alex Calderon</td>
<td>Lesson 1b - Radiation Fog</td>
<td>92</td>
<td>6/2/06</td>
</tr>
<tr>
<td><a href="mailto:alex.calderon@noaa.gov">alex.calderon@noaa.gov</a></td>
<td>Alex Calderon</td>
<td>Lesson 2a - Synoptic Weather Considerations: Forecasting Fog and Low Stratus</td>
<td>91</td>
<td>6/2/06</td>
</tr>
</tbody>
</table>
Appendix C: Sample php Code
Login Page

```php
<?php
session_start();

if ($_GET['quizStop'] != '') {
    $_SESSION['quizStop'] = $_GET['quizStop'];
    //echo "Value of quizStop is ".$_SESSION['quizStop'];
    //if a user backs all the way through a quiz after completing it to this page,
    // then all of the the session information is deleted. This is necessary to prevent
    // the user from then going forward and accidentally resubmitting the quiz by
    // using the forward button on the browser. The problem in that case is the answers
    // get deleted so the score comes back drastically lowered. This same code is in the
    // listQuizers.php file because depending on what path the user takes to get to the
    // question, it is necessary in both places.
    if ($_SESSION['emailCompleted'] == 'yes') {
        //unset $_SESSION['emailID'];
        unset($_SESSION['password']);
        unset($_SESSION['randKey']);
        unset($_SESSION['questions']);
        unset($_SESSION['answers']);
        unset($_SESSION['questionOptions']);
        unset($_SESSION['randAnswerOrder']);
        unset($_SESSION['optionArray']);
        unset($_SESSION['questionTypes']);
        unset($_SESSION['sequenceNumber']);
        unset($_SESSION['numQuestions']);
        unset($_SESSION['source']);
        //unset $_SESSION['quizNum'];
        //unset $_SESSION['quizID'];
        unset($_SESSION['randomQuestions']);
        unset($_SESSION['firstDone']);
        unset($_SESSION['lastDone']);
        unset($_SESSION['emailSent']);
        unset($_SESSION['emailCompleted']);
        //echo "Session_unset ".session_id();
    }
}
```

Page 73
<php
session_start();

if (!isset($_SESSION['quizPopUp']) || $_SESSION['quizPopUp'] == 'off') {
    //echo "value of quizpop-up is ", $_SESSION['quizPopUp'];
    //if a user backs all the way through a quiz after completing it to this page,
    //then all of the session information is deleted. This is necessary to prevent
    //the user from then going forward and accidentally resubmitting the quiz by
    //using the forward button on the browser. This problem is in that case in the answers
    //get deleted so the scorer comes back drastically lower. This same code is in the
    //lastQuizers.php file because depending on what path the user takes to get to the
    //questions, it is necessary in both places.
    if (!isset($_SESSION['emailCompleted']) || $_SESSION['emailCompleted'] == 'yes') {

        //unset($_SESSION['quizPopUp']);
        unset($_SESSION['emailCompleted']);
        unset($_SESSION['email']);
        unset($_SESSION['password']);
        unset($_SESSION['randomKey']);
        unset($_SESSION['questions']);
        unset($_SESSION['answers']);
        unset($_SESSION['questionOptions']);
        unset($_SESSION['randomAnswersOrder']);
        unset($_SESSION['answerArray']);
        unset($_SESSION['questionTypes']);
        unset($_SESSION['sequenceNumber']);
        unset($_SESSION['numQuestions']);
        unset($_SESSION['source']);
        unset($_SESSION['lastName']);
        unset($_SESSION['firstName']);
        unset($_SESSION['emailAddress']);
        unset($_SESSION['emailSent']);
        unset($_SESSION['emailCompleted']);
    }
</php>
User Account Page

```php
session_start();

if (!isset($_SESSION['quizID'])) {
    echo "Value of quizID is ".$_SESSION['quizID']."."
    //if a user backs all the way through a quiz after completing it to this page,
    //then all of the session information is deleted. This is necessary to prevent
    //the user from then going forward and accidentally resubmitting the quiz by
    //using the forward button on the browser. The problem in that case is the answers
    //get deleted so the score comes back drastically lower. This same code is in the
    //listQuizers.php file because depending on what path the user takes to get to the
    //quiz, it is necessary in both places.

    if ($_SESSION['emailCompleted'] == 'yes') {

        //reset: $_SESSION['quizID'] ;
        unset($_SESSION['quizID']);
        unset($_SESSION['password']);
        unset($_SESSION['randomKey']);
        unset($_SESSION['questions']);
        unset($_SESSION['answers']);
        unset($_SESSION['correctOptions']);
        unset($_SESSION['randomAnswersOrder']);
        unset($_SESSION['optionalArray']);
        unset($_SESSION['questionTypes']);
        unset($_SESSION['sequenceNumber']);
        unset($_SESSION['source']);
        unset($_SESSION['timeStamp']);
        unset($_SESSION['randomQuestions']);
        unset($_SESSION['firstUser']);
        unset($_SESSION['lastDate']);
        unset($_SESSION['emailSent']);
        unset($_SESSION['emailCompleted']);

        emailCompleted = 'emailCompleted'

    }

```
Quiz Results Page

```php
<?php
session_start();

if (isset($_SESSION['email']) && isset($_SESSION['password']) ) {
    $email = $_SESSION['email'];

    // Store answer in database
    require './DatabaseConnect.php';

    ?>

<html>
<head>
<title>NetEd User Registration</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<script language="JavaScript">
<link href="styles.css" rel="stylesheet" type="text/css"/>
</head>
<body>
<form name="quizResults">
<div align="center">
<br>
<br>
</div>
</form>
</body>
</html>
```
Reports Quiz Percentage

```php
session_start();

<!--

<?

<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>MyEd Registration and Quiz Reports</title>
</head>

<form action="/quizPercentagesReport.php" method="post">
  <img src="/images/header.jpg" width="700" height="38">
  <p align="center">Cost size="3">Get report for a quiz</p>
  <input type="submit" value="Submit"></form>

<?

1: (isset) [$_SESSION['valid_user'] ] )

//store answer in database
require './databaseConnect.php' ;

//fetch correct answers from the database for this quiz
$query = "select quiz_id, quiz_name from quizzes where posted_active = 'yes' order by quiz_name"
$result = mysql_query ( $query ) ;
$numResults = mysql_num_rows ( $result ) ;

<br />
<p align="center">
<form>
<input type="submit" value="Submit"></form>
</p>
```

Page 77
Appendix D: The Database Structure
Database Entry Point

Welcome to phpMyAdmin 2.5.4
MySQL 4.0.18 log running on localhost as dbmReg@localhost

MySQL
- Create new database: Documentation
- No Privileges
- Show processes: Documentation
- Databases
- Export
- Log out (*)

phpMyAdmin
- Language: English (en-us-8859-1)
- phpMyAdmin documentation
- Official phpMyAdmin Homepage
- ChangeLog | CVS | Lists

Query window
### Admin Table

**Database MetEdReg - Table admins running on localhost**

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Attributes</th>
<th>Null</th>
<th>Default</th>
<th>Extra</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_id</td>
<td>varchar(15)</td>
<td>No</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>password</td>
<td>varchar(10)</td>
<td>No</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Indexes:**

- No index defined

**Create an index on 1 columns**

**Row Statistic:**

- Rows: 20
- Row length: 180 Bytes
- Last update: Nov 8, 2004 at 18:04 PM

**Propose table structure [Documentation]**

Run SQL query/query sins on database MetEdReg [Documentation]

```sql
SELECT * FROM admins WHERE 1
```

**Fields:**

- user_id
- password
# Quiz Answers Table

![Database MetEdReg - Table answers running on localhost](image)

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Attributes</th>
<th>Null</th>
<th>Default</th>
<th>Extra</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>question_id</td>
<td>int(11)</td>
<td>No</td>
<td>No</td>
<td>auto_increment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>quiz_id</td>
<td>int(11)</td>
<td>No</td>
<td>No</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sequence_num</td>
<td>int(11)</td>
<td>No</td>
<td>No</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sub_sequence_num</td>
<td>int(4)</td>
<td>No</td>
<td>No</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>select_num</td>
<td>int(11)</td>
<td>No</td>
<td>No</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>select_text</td>
<td>mediumtext</td>
<td>No</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>answer</td>
<td>text</td>
<td>No</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>date_changed</td>
<td>timestamp(14)</td>
<td>Yes</td>
<td>WALK</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Indexes:**

<table>
<thead>
<tr>
<th>Keyname</th>
<th>Type</th>
<th>Cardinality</th>
<th>Action</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRIMARY</td>
<td>PRIMARY</td>
<td>6652</td>
<td></td>
<td>question_id</td>
</tr>
</tbody>
</table>

Create an index on 1 column.

**Space usage:**

- Data: 791,056 Bytes
- Index: 70,898 Bytes
- Total: 862,954 KB

**Row statistic:**

- Format: dynamic
- Rows: 8,288
- Row length: 144
- Row size: 154 Bytes
- Next Autoindex: 15,690
- Creation: Mar 30, 2005 at 11:48 AM
- Last update: May 30, 2005 at 9:38 AM

- **Print view**
- **Add new field:** 1 At End of Table
- **Propose table structure (Documentation)**

Run SQL query on database MetEdReg [Documentation]:

```sql
SELECT * FROM 'answers' WHERE 1
```

**Fields:**

- question_id
- quiz_id
- sequence_num
- sub_sequence_num
## Quiz Category Table

### Database MetEdReg - Table category_quiz running on localhost

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Attributes</th>
<th>Null</th>
<th>Default</th>
<th>Extra</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>category_name</td>
<td>varchar(100)</td>
<td>no</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>quiz_id</td>
<td>smallint(6)</td>
<td>no</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Indexes:** [Documentation]

No index defined

Create an index on 1 column

**Space usage:**
- Type: dynamic
- Rows: 125
- Row length: 30
- Row size: 39 Bytes
- Created: Jan 20, 2005 at 11:15 PM
- Last update: Nov 30, 2005 at 10:50 AM

**Row Statistic:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Usage</th>
<th>Statements</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Run SQL query/queries on database MetEdReg [Documentation]**

```
SELECT * FROM category_quiz WHERE 1
```

**Fields:**
- category_name
- quiz_id

---

Page 82
## Course Registration Table

### Database MetEdReg - Table courseRegs running on localhost

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Attributes</th>
<th>Null</th>
<th>Default</th>
<th>Extra</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>course_id</td>
<td>int(11)</td>
<td>No</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>quiz_id</td>
<td>int(11)</td>
<td>No</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>quiz_name</td>
<td>varchar(100)</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Indexes:**
- PRIMARY PRIMARY

**Field:** course_id, quiz_id

**Space usage:**
- Data: 5536 Bytes
- Index: 4103 Bytes
- Total: 9639 Bytes

**Row Statistic:**
- Rows: 0
- Row length: 62
- Row size: 100 Bytes

**Creation:** Jul 30, 2006 at 02:15 PM
**Last update:** Nov 09, 2006 at 12:07 AM

**SQL query:**
```
SELECT * FROM `courseRegs` WHERE 1
```

**Propose table structure [Documentation]:**

**Fields:**
- course_id
- quiz_id
- quiz_name

**Query window:**

**Compression:** None

**Browse:**

**Go:**
Courses Table

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Attributes</th>
<th>Null Default</th>
<th>Extra</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>course_id</td>
<td>int(8)</td>
<td>No</td>
<td>auto_incr</td>
<td></td>
<td></td>
</tr>
<tr>
<td>course_name</td>
<td>varchar(100)</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>courseNickname</td>
<td>varchar(30)</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>course_url</td>
<td>varchar(255)</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>direct_active</td>
<td>char(3)</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mixed_active</td>
<td>char(2)</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Indexes: [Documentation]

Create an index on 1 columns Go

Print view

Add new field: 1 At End of Table Go

Propose table structure [Documentation]

Run SQL queries on database MetEdReg [Documentation]

SELECT * FROM courses WHERE1

Rows: 12
Size: 119 Bytes
Next Autoindex: 42
Creation: Nov 29, 2005 at 11:19 AM
Last update: Nov 29, 2005 at 11:19 AM
### Course Completed Table

#### Database MetEdReg - Table coursesCompleted running on localhost

**Field** | **Type** | **Attributes** | **Null** | **Default** | **Extra** | **Action**
---|---|---|---|---|---|---
course_id | int[11] | Yes | 0 | | | |
email | varchar[100] | No | | | | |
date_completed | timestamp[14] | Yes | NULL | | | |

**Indexes**

Keyname | Type | Cardinality | Action | Field
---|---|---|---|---
course_id | UNIQUE | 505 | | course_id

Create an index on 1 columns

**Space usage**

<table>
<thead>
<tr>
<th>Type</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>30,738 Bytes</td>
</tr>
<tr>
<td>Index</td>
<td>36,624 Bytes</td>
</tr>
<tr>
<td>Total</td>
<td>47,362 Bytes</td>
</tr>
</tbody>
</table>

**Row Statistics**

<table>
<thead>
<tr>
<th>Statements</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format</td>
<td>dynamic</td>
</tr>
<tr>
<td>Rows</td>
<td>505</td>
</tr>
<tr>
<td>Row length</td>
<td>36</td>
</tr>
<tr>
<td>Row size</td>
<td>83 Bytes</td>
</tr>
<tr>
<td>Creation</td>
<td>Nov 03, 2005 at 11:42 AM</td>
</tr>
<tr>
<td>Last update</td>
<td>Nov 03, 2005 at 11:42 AM</td>
</tr>
</tbody>
</table>

- **Print view**
- **Add new field:** 1 | All End of Table | Go
- **Propose table structure:** [Documentation]

Run SQL query/queries on database MetEdReg [Documentation]

```
SELECT * FROM `coursesCompleted` WHERE 1
```

Insert

- **Show this query here again**
- **Location of the text file:**

Compression: | Autodetect | None | "gzipped" | "bzipped" | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

---

---

---
Question Attempts Table

![Database MetEdReg - Table quiz_attempts running on localhost](image)

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Attributes</th>
<th>Null</th>
<th>Default</th>
<th>Extra</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>email</td>
<td>varchar</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>date_submitted</td>
<td>timestamp</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Indexes: [Documentation]

Create an index on 1 columns

Space usage:

- Type: 276 KB
- Usage: 1 KB
- Total: 277 KB

Row statistic:

- Type: data
- Usage: 6,600
- Statements: 41
- Value: 42 Bytes

Print view

Add new field: 1

Propose table structure [Documentation]

Run SQL query on database MetEdReg [Documentation]

```sql
SELECT * FROM `quiz_attempts` WHERE 1
```

Fields:
- quiz_id
- email
- score
- date_submitted

Show this query here again
Or location of the textfile: [Browse]

Compression: None, "gzipped", "bzipped"
### Quizzes Table

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Attributes</th>
<th>Null</th>
<th>Default</th>
<th>Extra</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>quiz_id</td>
<td>int(11)</td>
<td>No</td>
<td>Yes</td>
<td>auto_increment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>quiz_name</td>
<td>varchar(150)</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>module_url</td>
<td>varchar(256)</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>category</td>
<td>varchar(256)</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>direct_active</td>
<td>char(3)</td>
<td>No</td>
<td></td>
<td>no</td>
<td></td>
<td></td>
</tr>
<tr>
<td>linked_active</td>
<td>char(3)</td>
<td>No</td>
<td></td>
<td>no</td>
<td></td>
<td></td>
</tr>
<tr>
<td>creator</td>
<td>varchar(50)</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>random_questions</td>
<td>char(3)</td>
<td>No</td>
<td></td>
<td>no</td>
<td></td>
<td></td>
</tr>
<tr>
<td>passing_score</td>
<td>tinyint(4)</td>
<td>No</td>
<td></td>
<td>D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>completed_quiz</td>
<td>int(11)</td>
<td>No</td>
<td></td>
<td>D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>data_changed</td>
<td>timestamp(14)</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Indexes:**

- PRIMARY
- PRIMARY
- quiz_name

**SQL Query:**

```sql
SELECT * FROM quizzes WHERE 1
```

**Fields:**

- quiz_id
- quiz_name
### Quiz Results Table

#### Database MetaEdReg - Table results running on localhost

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Attributes</th>
<th>Null</th>
<th>Default</th>
<th>Extra</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>quiz_id</td>
<td>int(11)</td>
<td>No</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email</td>
<td>varchar(100)</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>answers</td>
<td>text</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>score</td>
<td>smallint(3)</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>passed</td>
<td>char(3)</td>
<td>No</td>
<td>me</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>random_keys</td>
<td>varchar(55)</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>data_submitted</td>
<td>timestamp(14)</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Indexes: [Documentation]

- PRIMARY

Create an index on 1 column: [Go]

#### Space usage:

- Data: 1,020 KB
- Index: 380,038 Bytes
- Total: 2,192 KB

#### Raw Statistic:

- Statements: 9,294
- Raw size: 241 Bytes
- Creation: Nov 30, 2014 at 3:21 PM
- Last update: Nov 30, 2014 at 3:21 PM

---

*Print view*

*Add new field: 1* At End of Table [Go]

*Propose table structure [Documentation]*

Run SQL query/queries on database MetaEdReg [Documentation]

```sql
SELECT * FROM results WHERE 1
```

[Insert]
### Registered Users Table

#### Database MetEdReg - Table users running on localhost

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Attributes</th>
<th>Null</th>
<th>Default</th>
<th>Extra</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>email</td>
<td>varchar(100)</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>password</td>
<td>varchar(10)</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>reminder</td>
<td>varchar(30)</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>first_name</td>
<td>varchar(20)</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_name</td>
<td>varchar(20)</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>organization</td>
<td>varchar(50)</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>office</td>
<td>varchar(255)</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>supervisor_email</td>
<td>varchar(50)</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>address</td>
<td>varchar(30)</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>city</td>
<td>varchar(20)</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>state</td>
<td>char(2)</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>zip</td>
<td>varchar(6)</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>country</td>
<td>varchar(50)</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>privacy_policy</td>
<td>char(5)</td>
<td>No</td>
<td>no</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mail_list</td>
<td>char(5)</td>
<td>No</td>
<td>no</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>date_submitted</td>
<td>timestamp(14)</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>date_changed</td>
<td>timestamp(14)</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Indices:** [Documentation]

<table>
<thead>
<tr>
<th>Keyname</th>
<th>Type</th>
<th>Cardinality</th>
<th>Action</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>email</td>
<td>UNIQUE</td>
<td>2169</td>
<td></td>
<td>email</td>
</tr>
</tbody>
</table>

Create an index on 1 columns

**Space usage:**

- **Data:** 278,052 Bytes
- **Index:** 78,529 Bytes
- **Total:** 366,980 Bytes

**Raw size:** 165 Bytes

**Creation:** Nov 05, 2004 at 11:42 AM

**Last update:** Dec 01, 2004 at 08:33 AM

---

Page 90
**Browse on Quizzes Table**

This is an example view of data in the quizzes table in a mySQL database.
Manual Data Entry

Data can be manually entered into the database via this access point in mySQL.
SQL Testing

SQL can be tested from this point in the mySQL database.
MySQL Custom Search