2005

Web-based independent study program

Darryl Dwaine Scroggins

WEB-BASED INDEPENDENT STUDY PROGRAM

A Project

Presented to the

Faculty of

California State University,

San Bernardino

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Computer Science

---

by

Darryl Dwaine Scroggins

September 2005

WEB-BASED INDEPENDENT STUDY PROGRAM

A Project

Presented to the

Faculty of

California State University,

San Bernardino

---

by

Darryl Dwaine Scroggins

September 2005

Approved by:

[redacted]

Dr. David Turner, Chair, Computer Science

June 9, 2005

Date

[redacted]

Dr. George Georgiou

[redacted]

Dr. Kerstin Voigt

ABSTRACT

The Web-based Independent Study Program (WISP) is an on-line database program used to create and store educational records for members of Dikaios. Dikaios is a Christian educators association that offers an Independent Study Program (ISP). The WISP project will allow Dikaios home educators to create, store, edit, view, and print the forms and records that are required by Dikaios administrators and by the state of California. These forms are specifically the student's course of study, monthly evaluations, semester grade reports, graduation plan, and transcripts.

Using a database to store the classes that a student is planning to take will enable monthly evaluation reports and grade reports to be easily created and modified. It will also allow a high school graduation plan to be generated automatically from the data entered. The graduation plan shows the classwork that the student will take to meet graduation requirements. Having this plan available to the Dikaios administrators as well as to the home educator should allow any course requirement issues to be addressed early in the student's education.

It is the goal of this project to make the required planning and reporting simple enough so that it will no longer be a time consuming burden.

ACKNOWLEDGMENTS

I would like to thank the faculty of the Computer Science department for their patience while I was trying to decide on a project. I would also like to thank my graduate adviser, Dr. David Turner, who always seemed to believe that I would finish this project.

My wife and six children all deserve awards for tolerating seven years of graduate school. I do not take the sacrifices that they have made for granted. I know that I am truly a blessed man!

To Dikaios and their vision

of raising Godly children

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1 Background

Home educators are people who have undertaken the awesome and sometimes daunting task of educating their children at home. To enhance the home educating experience, or maybe just to survive it, many home educators choose to join a homeschool support group.

1.2 Independent Study Program

There are many reasons for joining a support group. Many groups offer the home educated child the opportunity to be involved in intramural sports, weekly cooperative classes, high school clubs, and other social events. Homeschool support groups that are recognized by the state of California as private schools also offer Independent Study Programs (ISP) to keep official records for the home educated student.

1.3 Homeschooling Records

The ISP is required by the State of California to keep certain records for each of its students. The records must show all of the courses being taken by each home educated student, their specific monthly progress in each class, their grades, and their daily attendance. It is the responsibility of the home educator to provide these

records to the ISP. Generating these records can be a tedious and time consuming task.

1.4 Project Goals

It is the goal of this project to make the planning and record keeping process as simple as possible for the home educator, specifically for Dikaios Christian Home Educators Association members. In addition, the reports generated by the system should meet the requirements specified in the Dikaios handbook [3] and should maintain the current report format. The content and layout of the reports should be the same as the paper forms currently in use.

CHAPTER TWO

SOFTWARE REQUIREMENTS SPECIFICATION

2.1 Purpose

The purpose of the Software Requirements Specification (SRS) is to specify the user's requirements of the system being designed. These requirements were derived by meeting with two Dikaios administrators and discussing the weaknesses associated with their existing record keeping system. The reporting requirements were derived from the Dikaios Christian Educators Association Handbook [3]. Additional input was solicited from end users as the project progressed.

2.2 Project Overview

The Web-based Independent Study Program (WISP) provides a solution for home educators and ISP administrators. It also contains components that will be used by home educators and their students.

2.3 Business Context

Dikaios Christian Educators Association is currently using several computer programs along with manual record keeping techniques to administer the educational records of approximately 200 Junior High and High School students. This process is tedious and time consuming.

2.4 Project Functions

The primary functional requirements for WISP are to:

- Store and process member information such as: name, home address, phone number, email address.
- Keep track of each student's attendance records.
- Keep track of each student's course of study.
- Generate student report cards and transcripts.
- Facilitate communication between administrators and Home Educators.
- Allow an on-line learning program to be linked to it.
- Provide a back-up and restore procedure.

2.5 Similar Systems

There are web-based school administration programs, such as K12admin [7], commercially available. These programs use the large class model and do not address the unique requirements of the home educator.

The home educator needs a system that allows them to create custom educational plans for each of their children and track their progress monthly.

2.6 User Characteristics and Use Cases

There will be two types of users of the WISP system; the home educator and the system administrator. The home educator will use the system for a variety of planning and reporting tasks. The use case diagram in Figure 1 shows that the home educator can use the system to create or view

courses of study, monthly evaluations, and grade reports for each of their students. The home educator can choose to display a graduation plan, which summarizes all of the course of studies planned for a single student. The last use of the system is to display a transcript which shows the students educational achievements.

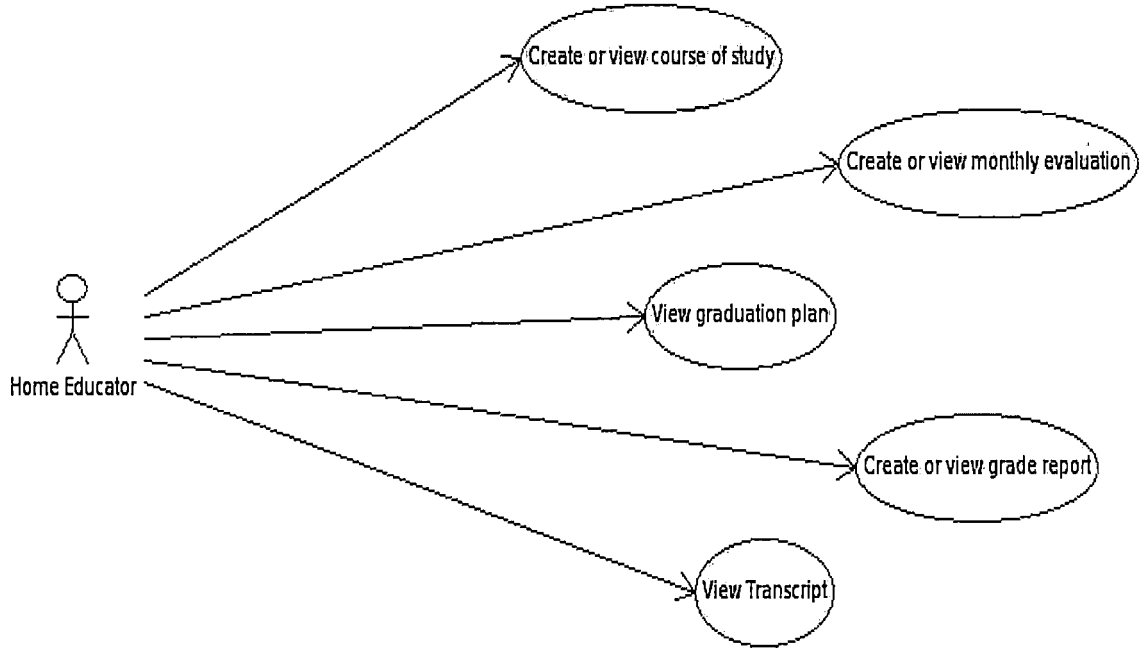


Figure 1. Home Educator Use Cases

The second type of user is the ISP administrator. (See Figure 2.) This user will have the ability to add, modify, and list member information, create mailing labels, and access all information stored in the WISP database. The administrator is responsible for making hard copies and archiving the records created by the home educator.

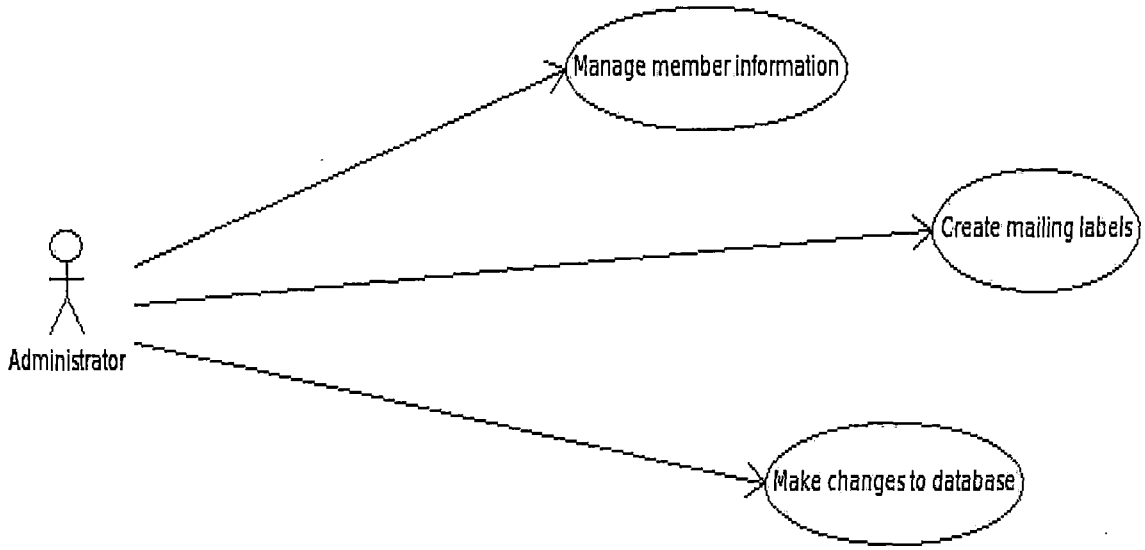


Figure 2. Administrator Use Cases

2.7 General Constraints

This project was created using open source software. To facilitate support and future enhancements, commonly used languages and platforms were used. The WISP system will be on-line by August 1st, 2005.

2.8 Functional Requirements

Functional requirements are what the system must accomplish. Other types of requirements, such as interface requirements or performance requirements, describe how the system accomplishes its functional requirements and are dealt with later in sections 2.9 and 2.10.

This section lists the functional requirements in order of importance. Each functional requirement is first described and then each of the following issues are

addressed: its criticality to the project, any technical issues associated with it, its costs and effect on the schedule, the risks associated with it, and the final result.

2.8.1 Generate Student Transcripts

2.8.1.1 Description. Generating a transcript is currently a very time consuming process. It requires the tabulation of several pages of grades, the classification of courses completed by description, the addition of attendance records, and finally the merging of all the information onto a complex form. To accomplish this task each student's course of study, their grades, and their attendance must be recorded. Each of these separate tasks are functional requirements of the system.

2.8.1.2 Significance. This function, along with its supporting sub-functions, will require about 60 percent of the project development time. It is an essential system function that must work for the project to be considered a success.

2.8.1.3 Technical Issues. To meet this requirement the system must store student information in a manner that will allow it to be tabulated as required on the transcript. The program must also be able to determine which graduation requirements are met by which courses. The current transcript being created is a single page.

This may need to be changed to allow for variation between student records.

2.8.1.4 Risks. If under some circumstance a transcript cannot be formated properly it should be output to a file that can be edited by hand.

2.8.1.5 Final Outcome. The WISP system can generate a preliminary transcript that shows attendance, credits, and grades. (See Appendix D.4.) The preliminary transcript will be useful to the home educator but it is not in a format that can be used as a final transcript.

2.8.2 Facilitate Communication

2.8.2.1 Description. There are many news worthy events that the membership can be made aware of either through a web page or by email.

2.8.2.2 Significance. This function should not take more than 5 to 10 percent of the development time. It is not a critical function but it should come at relatively little cost.

2.8.2.3 Technical Issues. Allowing certain Home Educators to publish to the web simply and securely may be a challenge. Setting up a system to email the membership automatically may also be a bit of a challenge.

2.8.2.4 Risks. This method of communication will be in addition to all of the current methods. If this feature does not get implemented it will be considered lost opportunity, but the project may still be a success.

2.8.2.5 Final Outcome. This requirement was met by installing and configuring Mambo [9] as a content management system. It can be accessed at http://www.dikaios.org/mambo/.

2.8.3 Integrate a Course Management System

2.8.3.1 Description. There are several open source projects that have the potential to be incorporated with the WISP program. Adding one of these programs would allow a co-op teacher to post information such as syllabus, assignments, class news, etc. to an area designated as the class website. Students and parents would be able to read this material and correspond with the instructor.

2.8.3.2 Significance. This function should have very little impact on the schedule. It should be a matter of finding the software and configuring it. Some time may be used trying to integrate the software. This feature is not critical to the success of this project. If it is included it will be a nice feature for those involved in the co-op.

2.8.3.3 Technical Issues. It will take time to find a system that can be used in conjunction with the WISP system. The system will need to be installed and configured.

2.8.3.4 Risks. Since this software should be very loosely coupled with the main system it should pose no risk to the success of the project.

2.8.3.5 Final Outcome. This requirement was met by installing and configuring Moodle [10] as a course management system. It can be accessed at http://www.dikaios.org/moodle/.

2.9 Interface Requirements

This section describes how the software interfaces with other software products or users for input or output.

2.9.1 User Interfaces

The WISP interface will be a standard web browser such as Firefox or Internet Explorer.

2.9.2 Hardware Interfaces

The WISP server will not require any significant hardware interfaces. Clients will access the server either through an ethernet card or a dial-up connection depending on the configuration.

2.9.3 Communications Interfaces

WISP will be designed to run as an intranet or internet program. The program will use HTTPS and TCP/IP to communicate securely between clients and the server.

2.9.4 Software Interfaces

To insure that forms and labels have a consistent look WISP may make use of data formats such as portable document format (PDF). Data will be sent to the clients in the appropriate format preceded by Mime headers that will cause an appropriate external program to load and interpret it.

2.10 Performance Requirements

The WISP system will meet all performance expectations when run on a modest personal computer (1Ghz 586 or better).

2.11 Design Constraints

WISP should not require any proprietary software to run effectively.

2.12 Other Non-functional Attributes

2.12.1 Security

Security is always a concern when storing personal information. Data will be transferred between client and server using secure socket layer (SSL). User access to the system will be limited by requiring a valid email address and password combination. Administrator access will be restricted using "htaccess". All passwords will be encrypted before they are stored in database.

2.12.2 Backup and Recovery

The ability to recover from data loss or system crash is essential. A back-up plan and installation instructions will be created.

2.12.3 Re-usability

Re-usability is not a mandatory requirement but is deemed a very desirable goal. This should be considered when making design decisions specific to Dikaios. At a minimum, all site specific constants and logos should be set in a single initialization file.

2.13 Project Schedule

The following tasks were implemented in sequential order:

- Created a working prototype of all system reports using the database and scripting language on an existing server.
- Obtained user feedback on the report formats and the data entry process.
- Fine tuned the reports and added the required programs for editing data.
- Integrated the on-line learning system.
- Allowed the system to be used for a few weeks and obtained feedback.
- Made changes based on user feedback.

2.14 Budget

There was no budget allocated for this project. All necessary hardware existed and all necessary software was freely available so the lack of finances did not cause a problem.

CHAPTER THREE

PROJECT DESIGN

3.1 System Architecture

The WISP project was created using what is commonly referred to as the LAMP software infra-structure. LAMP is an acronym for Linux, Apache [1], MySQL [11], and PHP [14]. Each of these four components of LAMP are extremely successful examples of open source projects. See references for the web address of each of their main project pages.

All user interaction with the WISP system occurs by way of a web browser interface via the internet. This allows the system to be easily accessed by each of Dikaios' home educators. See Figure 3.

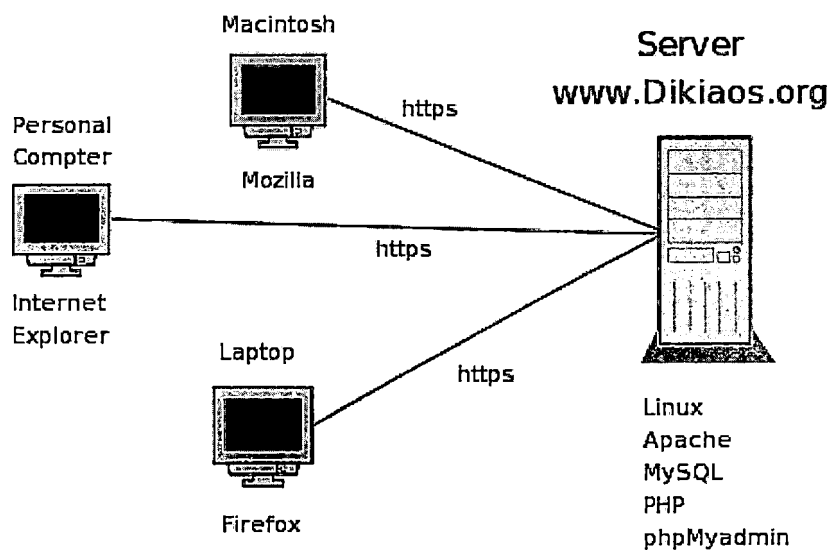


Figure 3. Client / Server Architecture

3.2 System Interface

The WISP system uses a web browser to provide the graphical user interface between the user and the MySQL database system.

3.3 General User Tasks

The home educator can use the system to create or view a course of study for their students. Once a course of study is created for a given semester, monthly evaluations and grade reports can be created. A graduation plan, which summarizes all of the course of studies for a single student, can be generated by the system. The system will also let the home educator display a preliminary transcript.

3.3.1 Create Course of Study

Once the user is logged in, they will be presented with a list of tasks. (See Figure 4.)

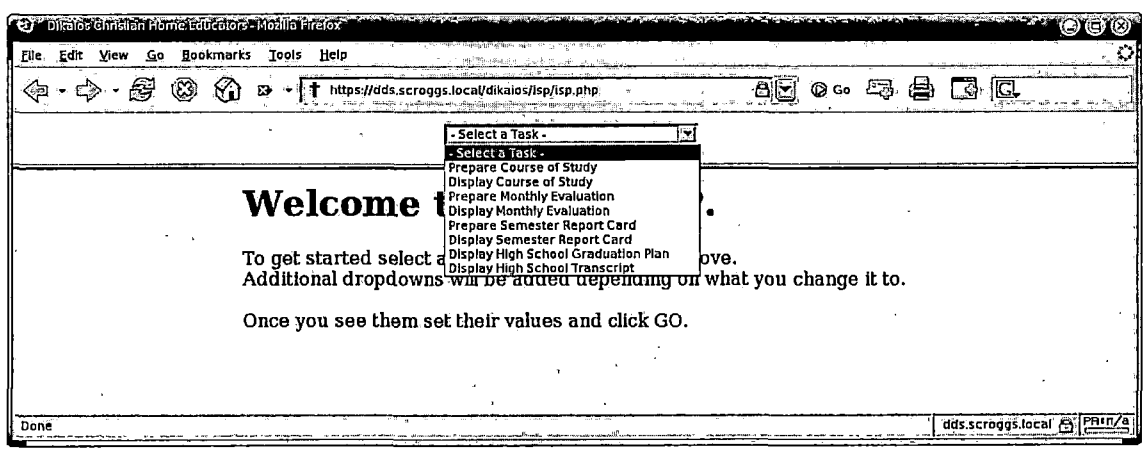


Figure 4. Select a Task

One of the first tasks that should be performed is to create a course of study for each of the students being home educated. (See Figure 5.) This process involves selecting a student name, a grade, and a semester and then clicking GO. Classes are added to the course of study one at a time. First a subject area is selected and then a specific class from within the selected subject area is chosen. Lastly all of the class details are input and the data is saved. This process is repeated until all of the classes for the given semester are entered.

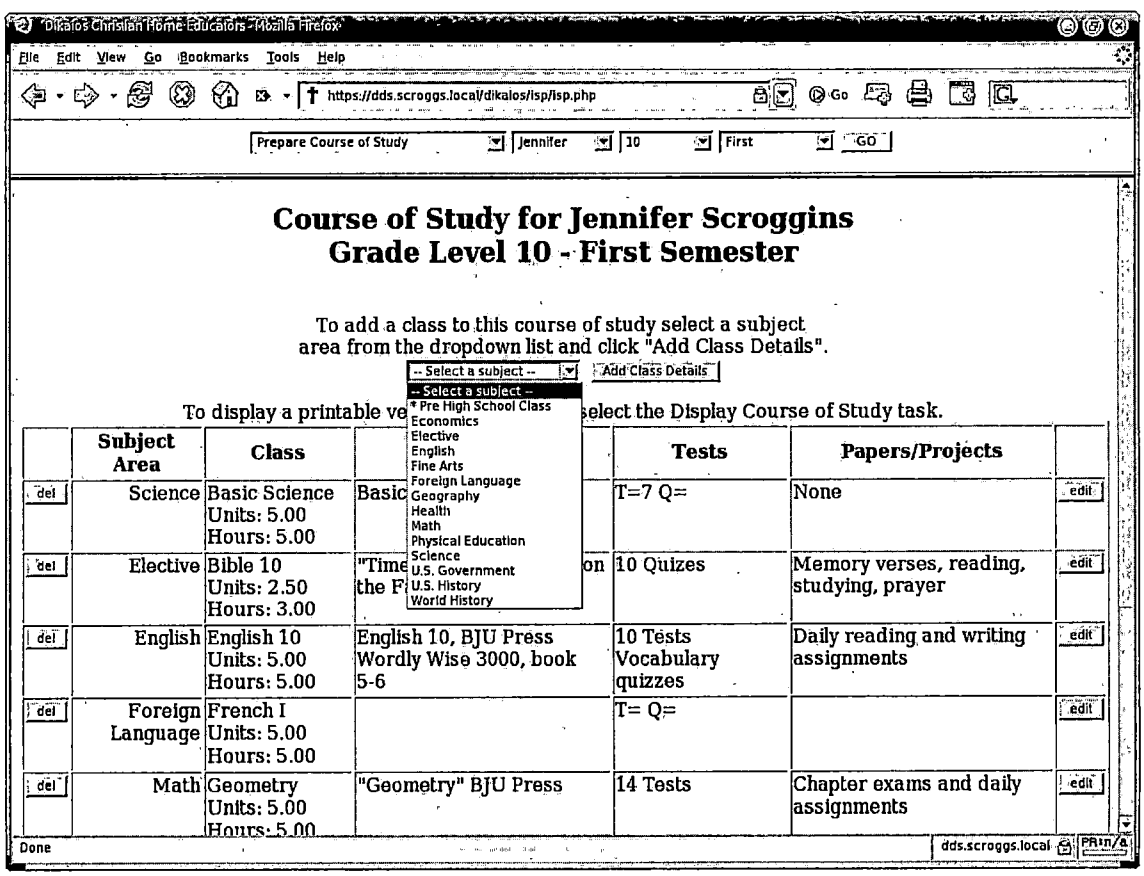
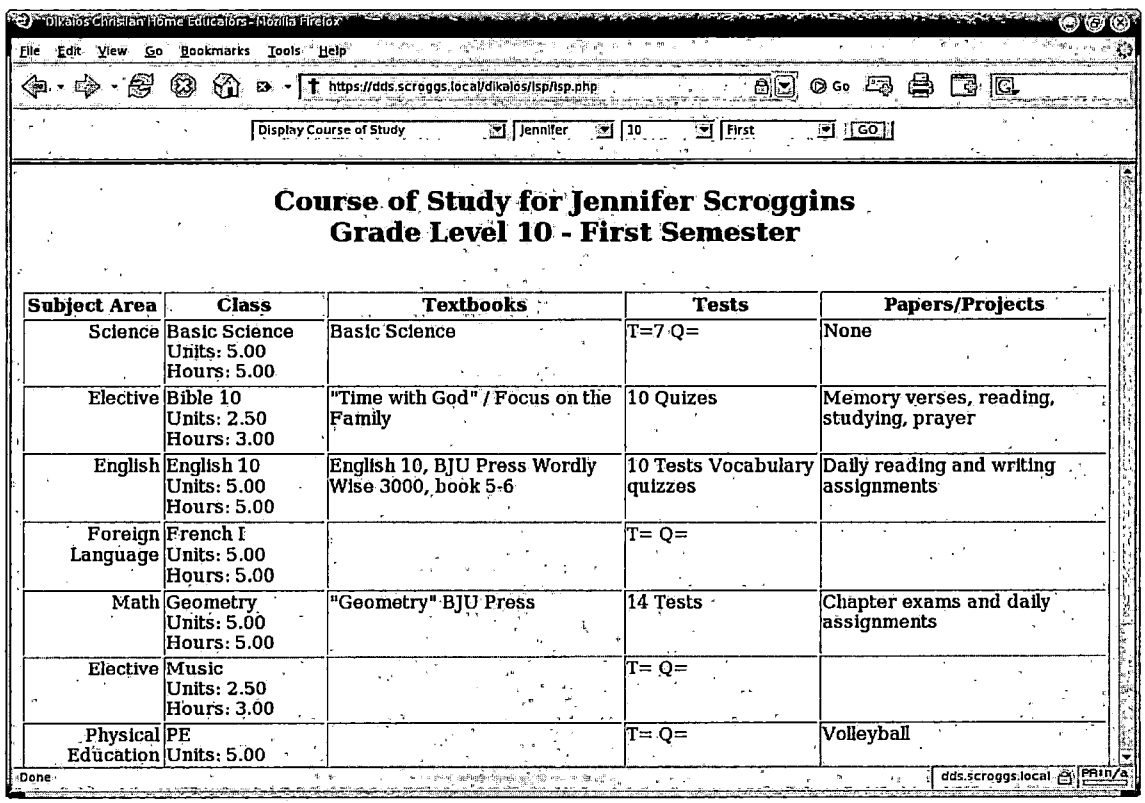


Figure 5. Prepare Course of Study

3.3.2 View Course of Study

The course of study can be viewed at any time by selecting the View course of study task from the select menu. The results will look like Figure 6. (See Appendix A.1 for a sample of the printed output.)



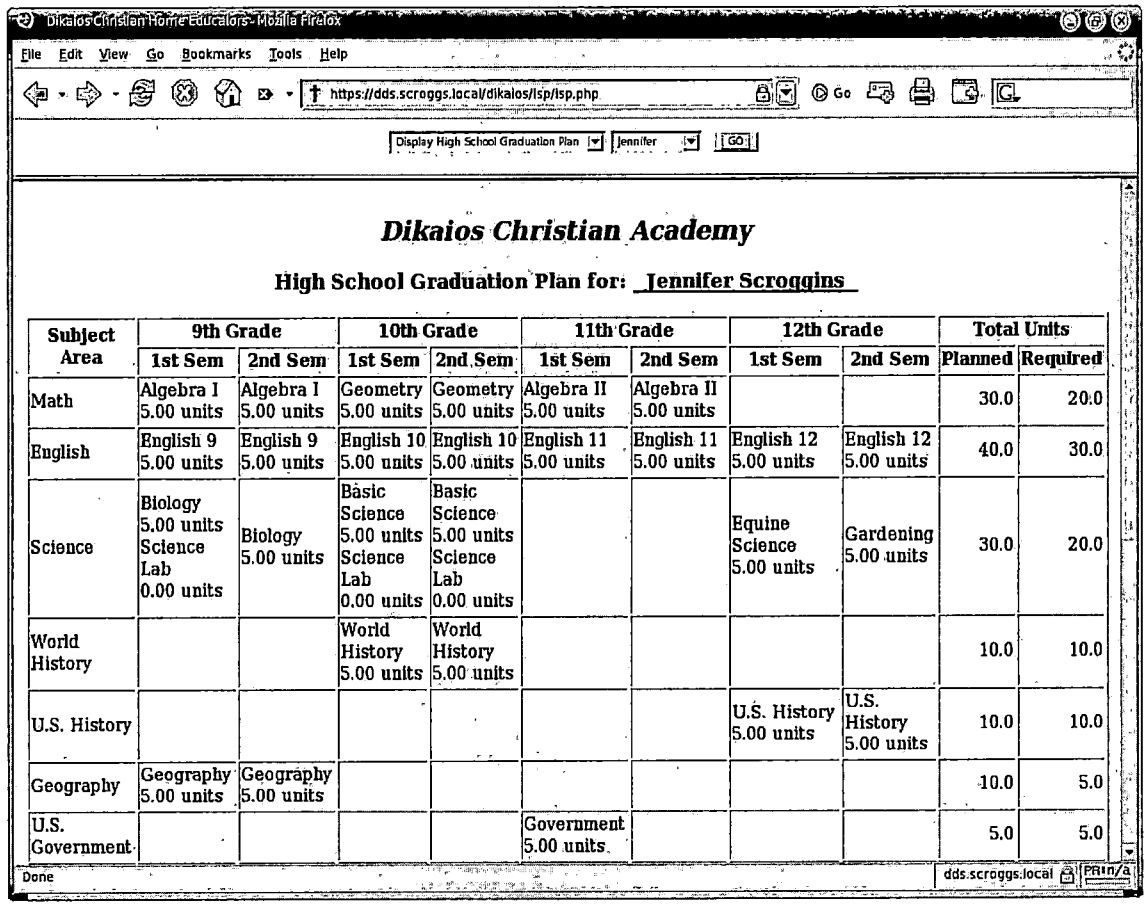
Subject Area	Class	Textbooks	Tests	Papers/Projects
Science	Basic Science Units: 5.00 Hours: 5.00	Basic Science	T=7 Q=	None
Elective	Bible 10 Units: 2.50 Hours: 3.00	"Time with God" / Focus on the Family	10 Quizes	Memory verses, reading, studying, prayer
English	English 10 Units: 5.00 Hours: 5.00	English 10, BJU Press Wordly Wise 3000, book 5-6	10 Tests Vocabulary quizzes	Daily reading and writing assignments
Foreign Language	French I Units: 5.00 Hours: 5.00		T= Q=	
Math	Geometry Units: 5.00 Hours: 5.00	"Geometry" BJU Press	14 Tests	Chapter exams and daily assignments
Elective	Music Units: 2.50 Hours: 3.00		T= Q=	
Physical Education	PE Units: 5.00		T= Q=	Volleyball

Figure 6. View Course of Study

3.3.3 Display Graduation Plan

The graduation plan is a summary of all of the course of studies that have been entered into the system for a selected student. (See Figure 7.) It shows when each of the classes required for graduation will be taken and

provides a quick verification that all of the unit requirements will be met. (See sample output in Appendix A.3 for a complete graduation plan.)



Dikaios Christian Academy

High School Graduation Plan for: Jennifer Scroggins

Subject Area	9th Grade		10th Grade		11th Grade		12th Grade		Total Units	
	1st Sem	2nd Sem	1st Sem	2nd Sem	1st Sem	2nd Sem	1st Sem	2nd Sem	Planned	Required
Math	Algebra I 5.00 units	Algebra I 5.00 units	Geometry 5.00 units	Geometry 5.00 units	Algebra II 5.00 units	Algebra II 5.00 units			30.0	20.0
English	English 9 5.00 units	English 9 5.00 units	English 10 5.00 units	English 10 5.00 units	English 11 5.00 units	English 11 5.00 units	English 12 5.00 units	English 12 5.00 units	40.0	30.0
Science	Biology 5.00 units Science Lab 0.00 units	Biology 5.00 units	Basic Science 5.00 units Science Lab 0.00 units	Basic Science 5.00 units Science Lab 0.00 units			Equine Science 5.00 units	Gardening 5.00 units	30.0	20.0
World History			World History 5.00 units	World History 5.00 units					10.0	10.0
U.S. History							U.S. History 5.00 units	U.S. History 5.00 units	10.0	10.0
Geography	Geography 5.00 units	Geography 5.00 units							10.0	5.0
U.S. Government					Government 5.00 units				5.0	5.0

Figure 7. Display Graduation Plan

3.3.4 Prepare Monthly Evaluation

Another task that can be preformed using WISP is the required monthly evaluations. (See Figure 8 and Figure 9.) This involves logging attendance for the month, pages read, test grades, and any other notes of interest. Being able to create this form and re-edit it should prove to be a

great help for the home educator. (See Appendix A.2 for a sample of the printed output.)

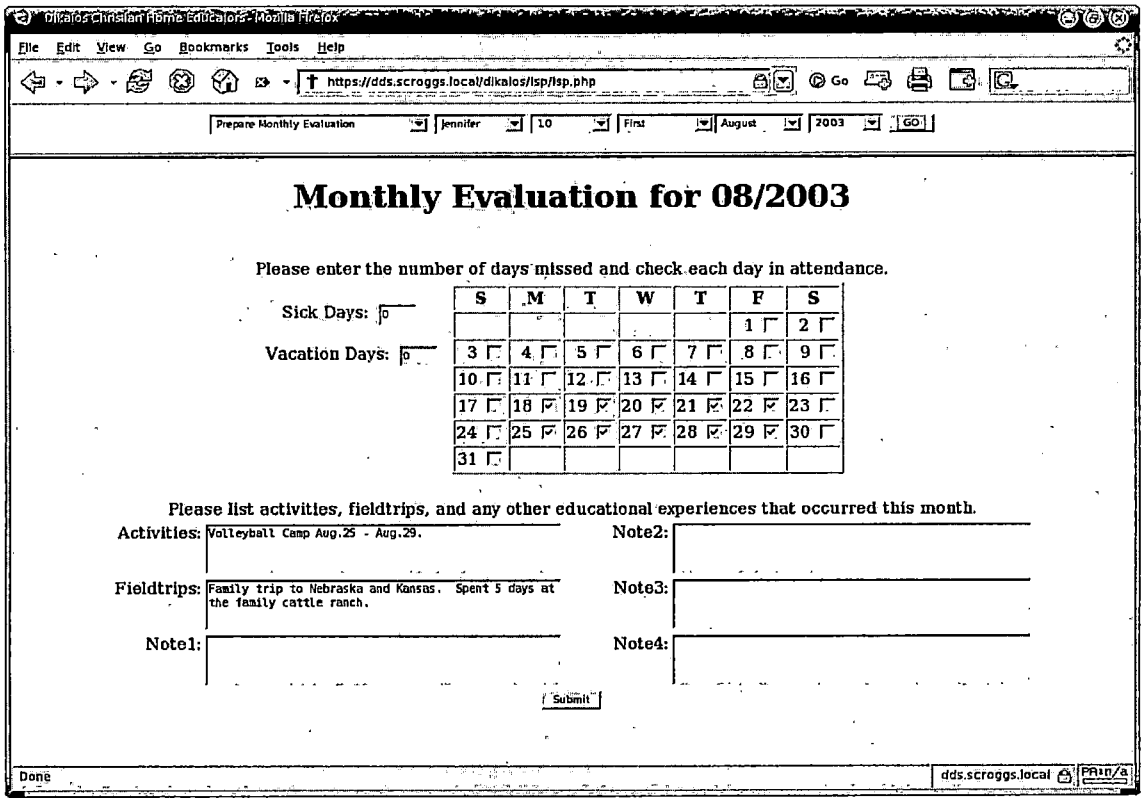


Figure 8. Prepare Monthly Evaluation Part 1

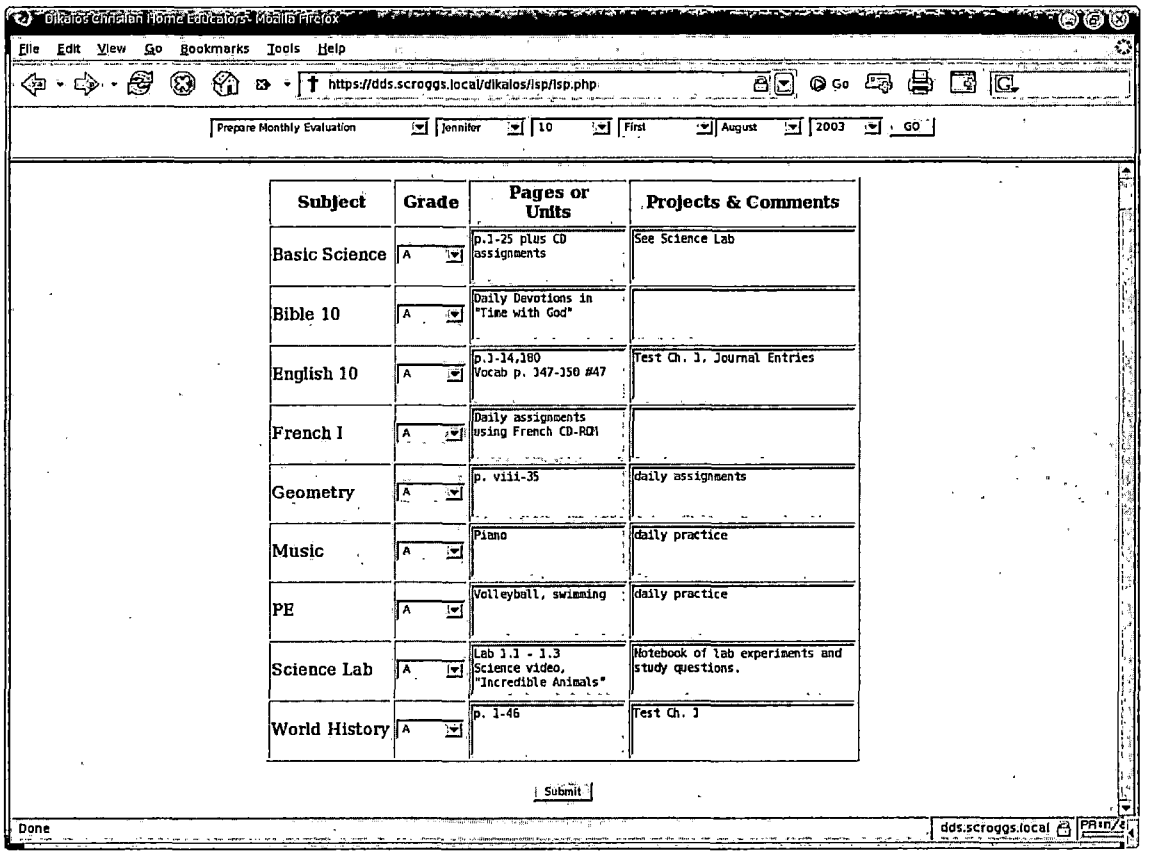


Figure 9. Prepare Monthly Evaluation Part 2

3.3.5 Prepare Semester Report Cards

A similar task that occurs at the end of each semester is the preparation of report cards. The grades given on the report cards are used to determine the student's grade point average (GPA) and they are the grades that appear on their transcript. (See Figure 10, Figure 11, and Appendix A.4)

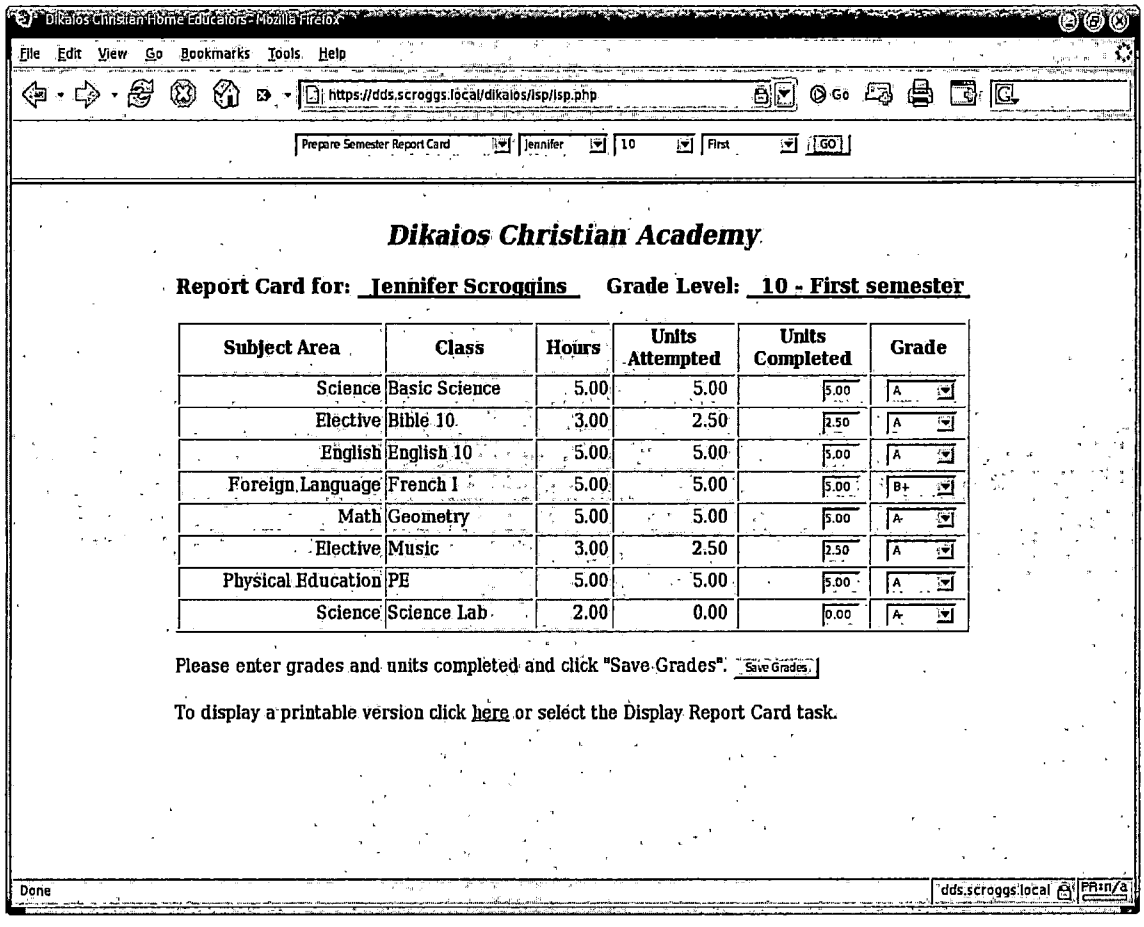


Figure 10. Prepare Semester Report Cards

3.3.6 Display High School Transcript

This task will generate a report that shows attendance, units completed versus units required, grades, extra-curricular activities, and notes, for a student's high school years. The preliminary transcript can be used to keep track of a student's progress towards graduation. (See Figure 11 and Appendix A.5.)

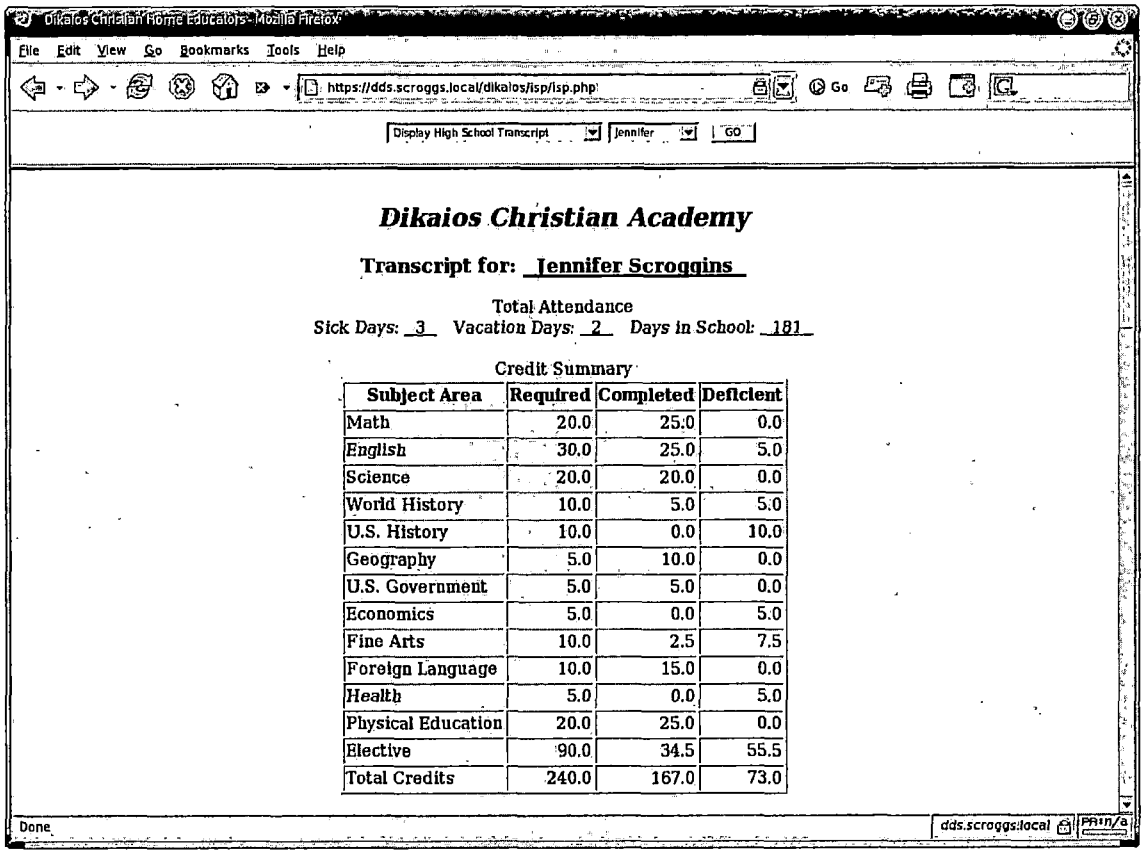


Figure 11. Display Transcript

3.4 Administrator Tasks

The ISP administrators have the ability to add and edit member information, print mailing labels, and make changes to database as necessary. To log in as an administrator a user must access the admin directory and provide proper authentication. The admin directory is protected using htaccess. (See Setting up Administrative Security on page 44.)

Once the administrator is logged in, the main administrative menu is displayed. (See Figure 12.)

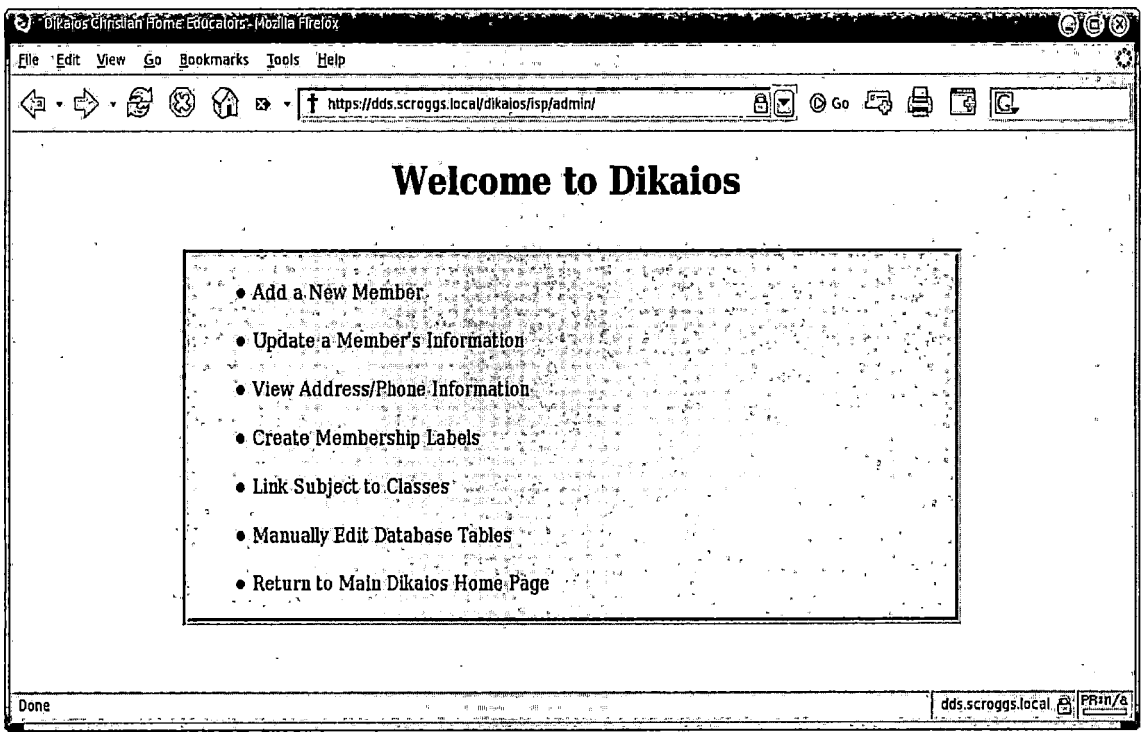


Figure 12. Administrative Menu

3.4.1 Manage Member Information

The ISP administrator is responsible for insuring that all member information is correctly entered in the system. This can be accomplished using the Add and Update Member Information Tasks. (See Figure 13.)

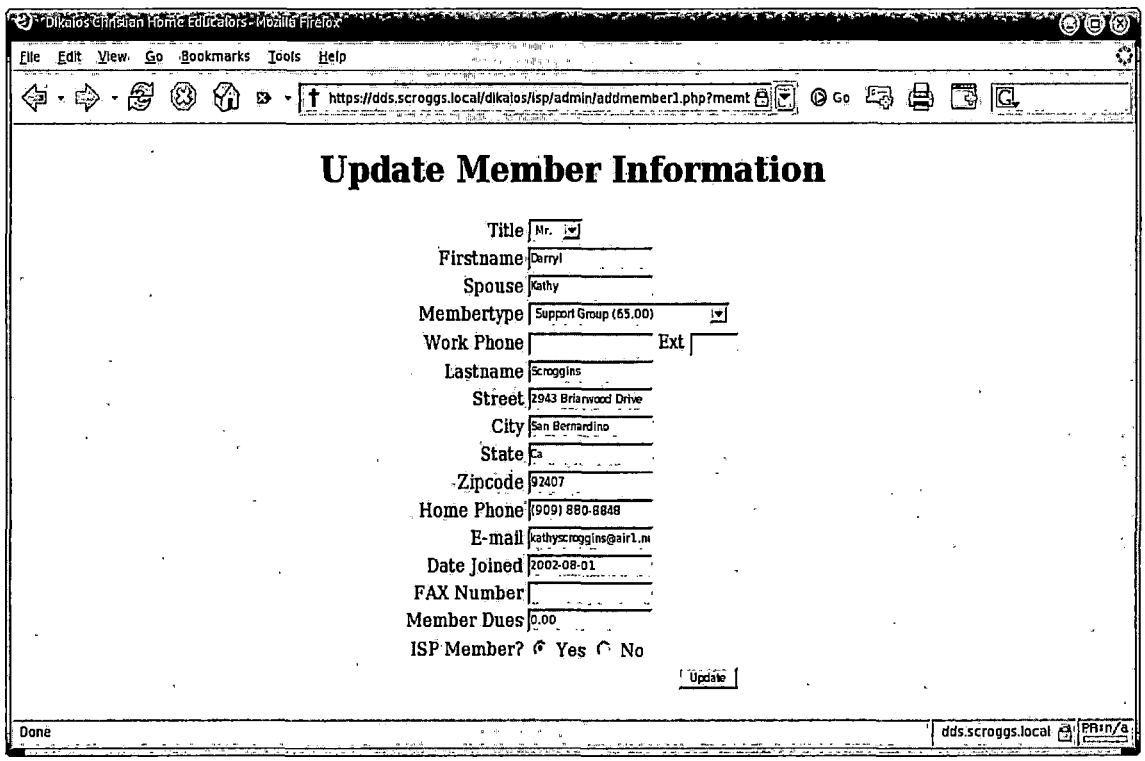


Figure 13. Update Member Information

The administrator can create lists of member information by using the "View Address / Phone" task. The system prompts for a name to look up. The name can be entered completely to display an individual, partially to display a group of people, or it can be left blank to display all members. The information displayed can also be controlled by selecting the check boxes: phone, address, or email. (See Figure 14 and Figure 15.)

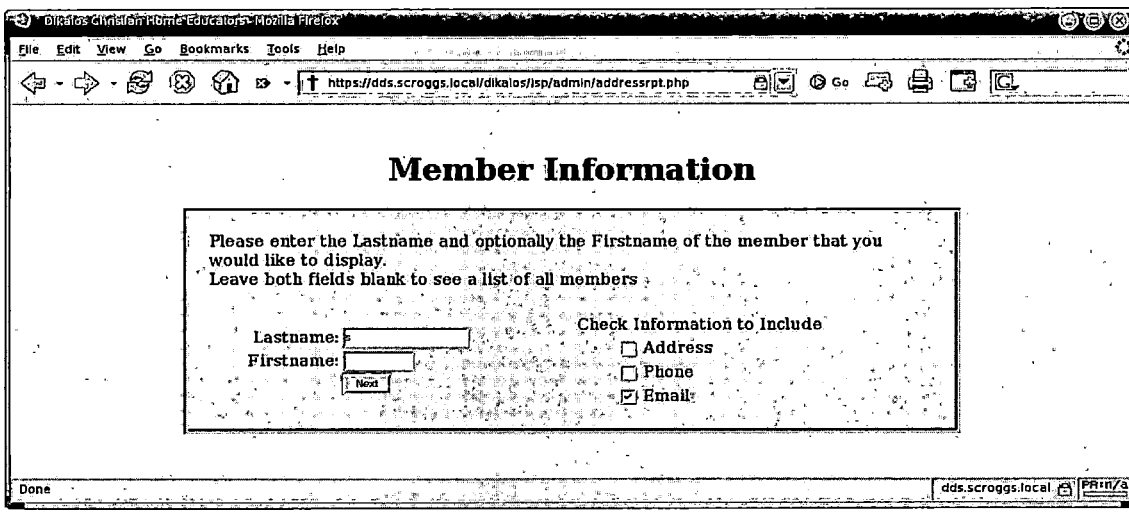


Figure 14. Select Member Information

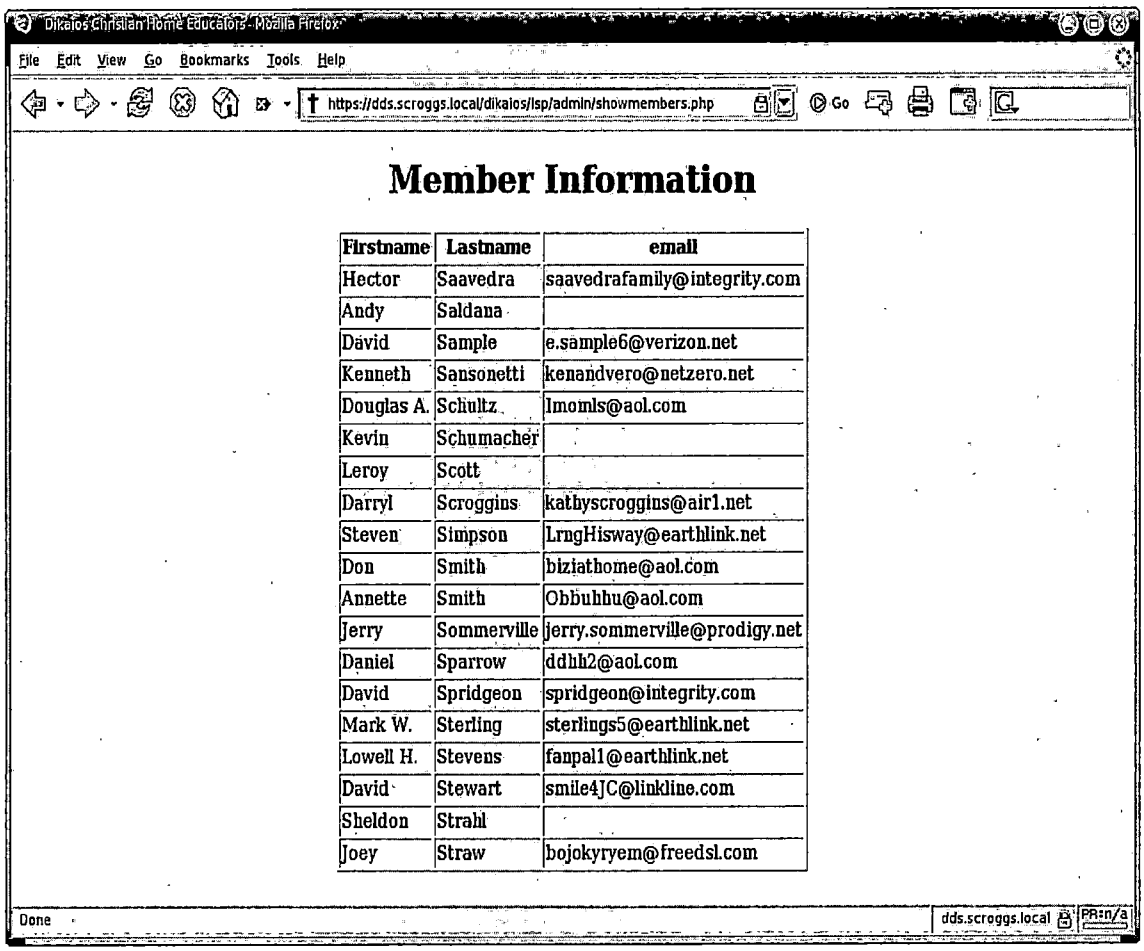


Figure 15. Member Information

3.4.2 Create Mailing Labels

The WISP system has the ability to create a file that can be used to print mailing labels. The file is created in portable document format (PDF) and requires Adobe Acrobat or a similar program to print. (See Figure 16.)

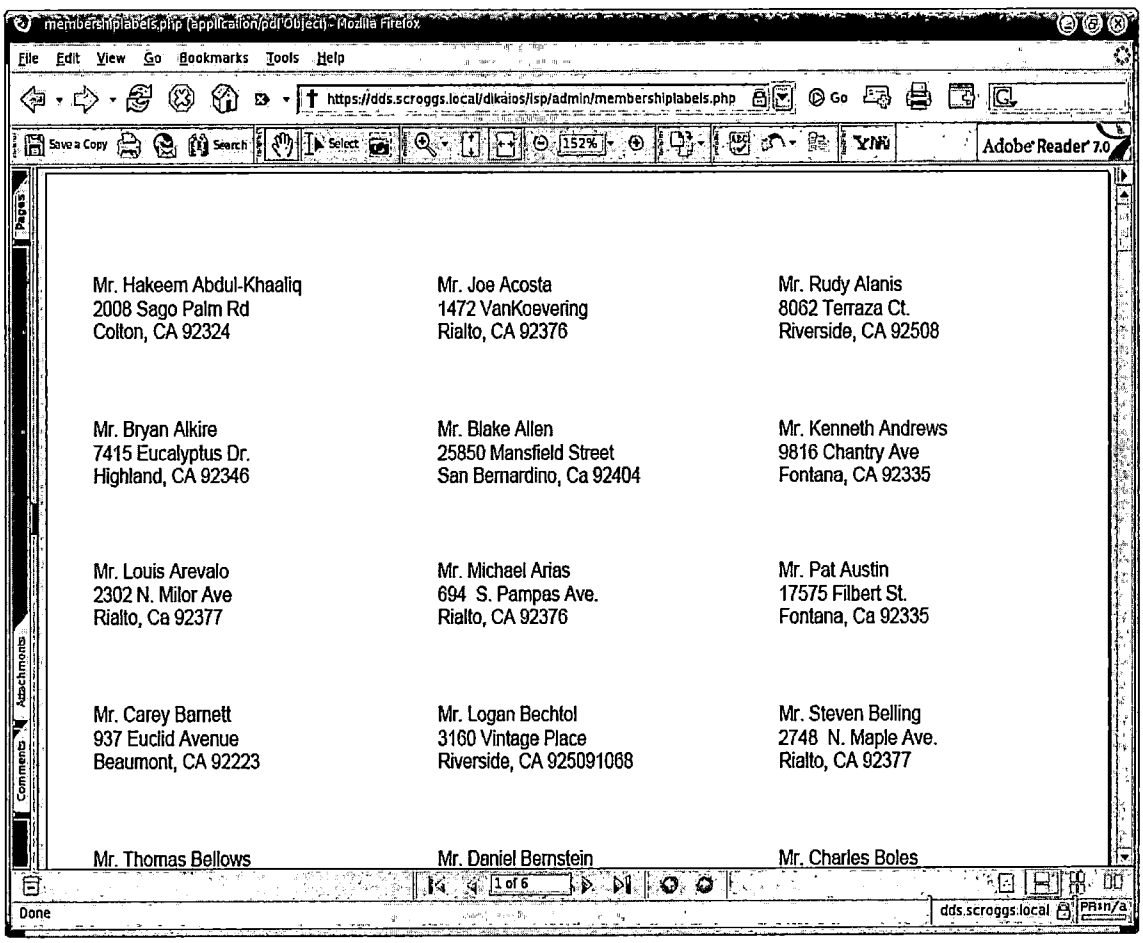


Figure 16. Create Mailing Labels

3.4.3 Complete Database Access

The ISP administrator also has the ability make changes to the database. To link the class to the appropriate subject the Link Subjects to Classes task can be used. The subject area is selected in the first screen. (See Figure 17.)

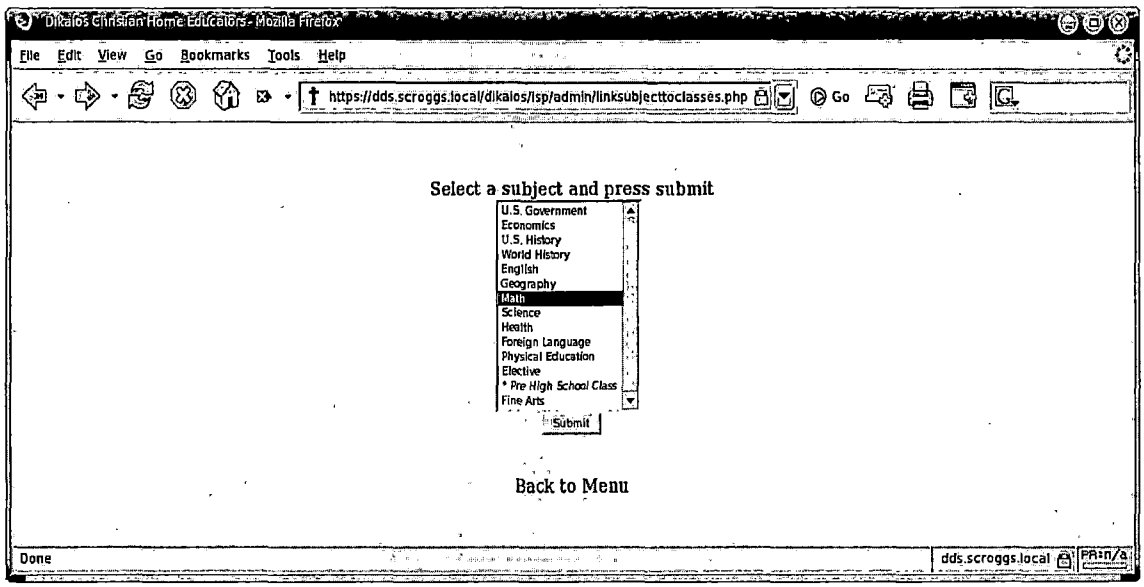


Figure 17. Select a Subject

Once the subject has been selected all of the classes that meet the requirements for that subject area are selected on the second screen. (See Figure 18.)

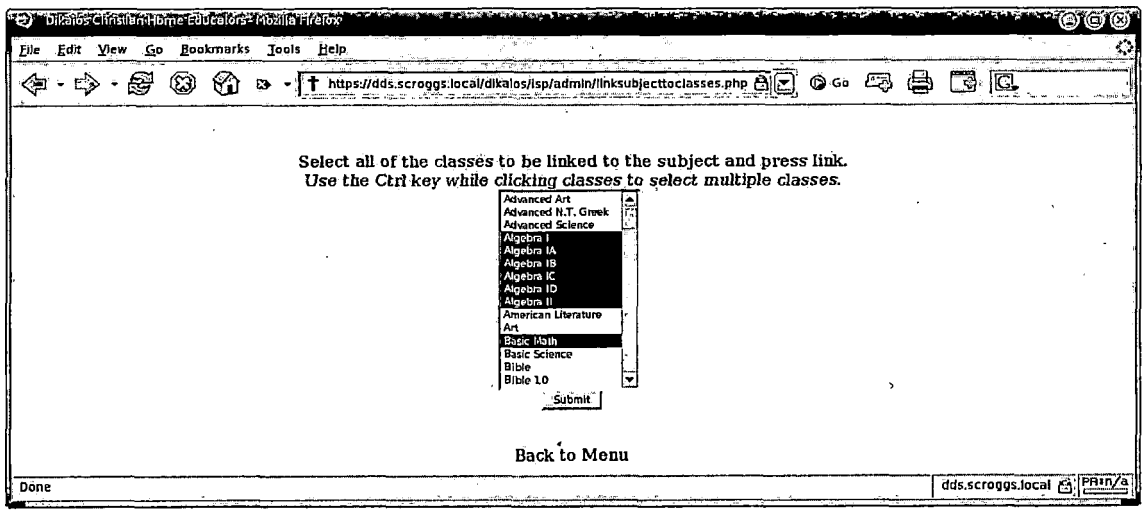


Figure 18. Link Classes to Subjects

There is one last option on the Administrator's Menu which allows the database to be manually edited. This feature is needed to change information that does not currently have a WISP form interface. An example of this is the number of units required in each subject area. This feature is implemented by using phpMyAdmin [15]. (See Figure 19.) Ideally there would be no need for this feature but having it will allow the ISP administrator to make the necessary changes to keep the system usable.

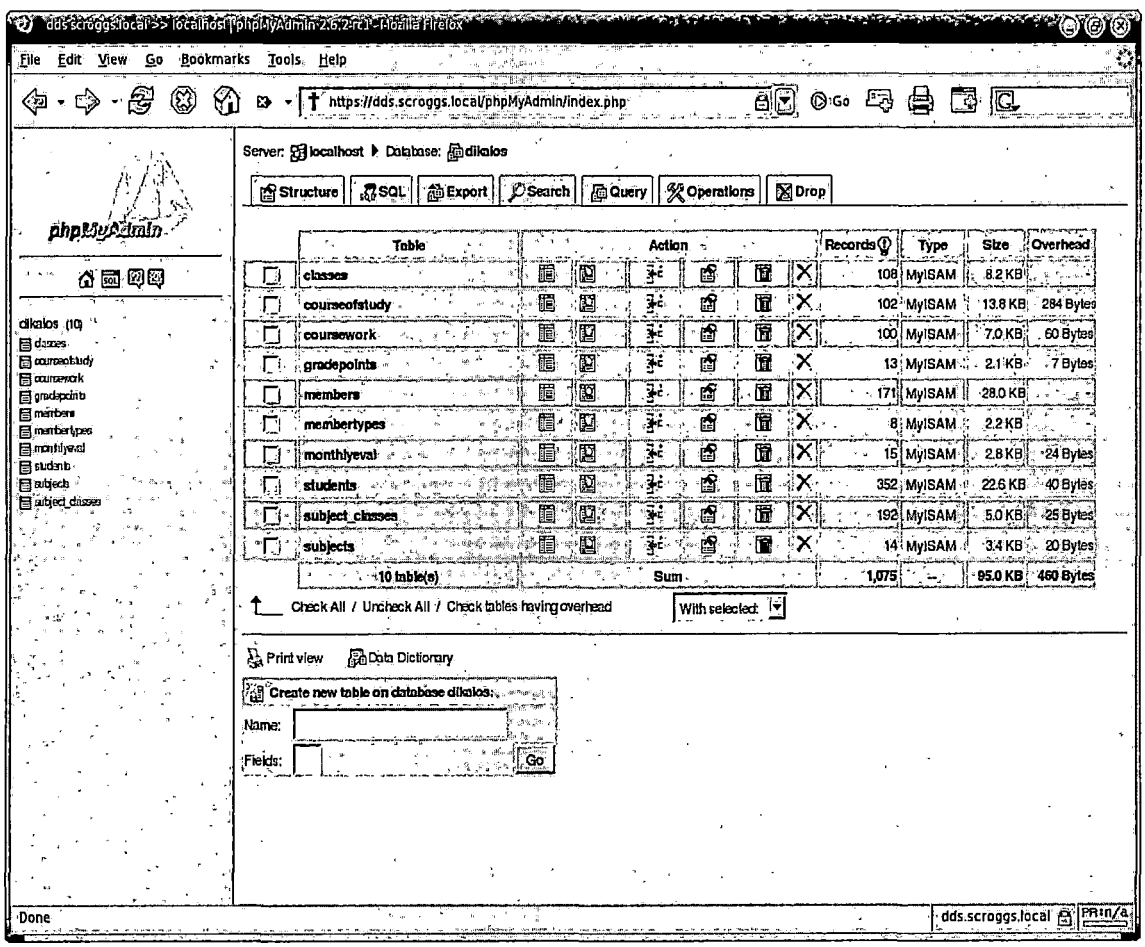


Figure 19. The phpMyAdmin Program

3.5 Software Architecture

The main WISP project files make extensive use of PHP scripting [5,6,8] and relatively little use of HTML formatting. Each of the main files has the same general page structure. (See Figure 20.)

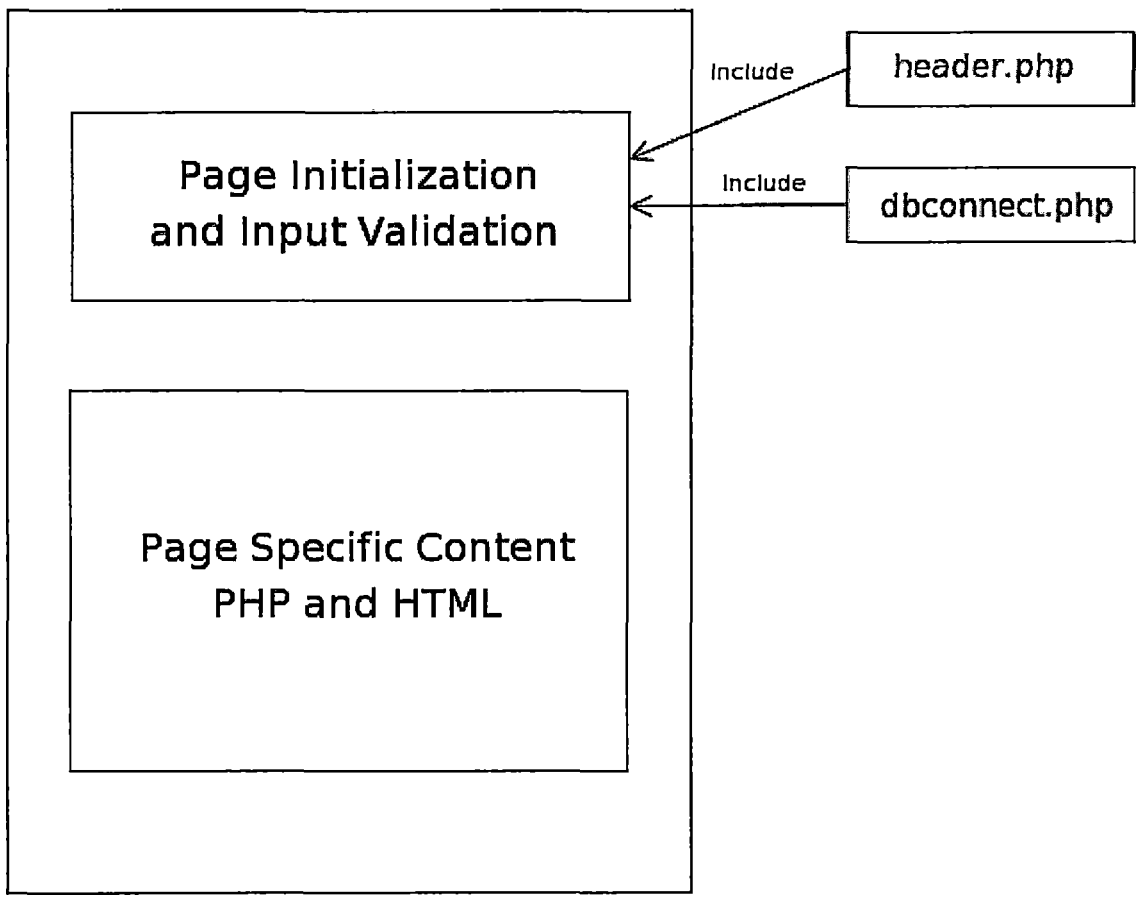


Figure 20. General Page Structure

First the header.inc file is pulled in by use of an include command. The file contains some header meta tags, a default page title, and some cascading style sheet (CSS) formatting [4]. Next the the variables that are expected

to be passed in are tested to verify that they are set. If the page accesses the database, the file dbconnect.php is included, and the function dbconnect is called to make the connection to the database.

The project files can be grouped into four categories; frame layout and the menu system, user authentication and password management, data entry and reporting, and system administrative tasks.

3.5.1 Frame Layout and Menu System

The file isp.php contains the main frame set definition. It defines a top frame containing the task menu and a bottom frame which initially contains a welcome message and some brief instructions regarding selecting tasks from the task menu.

The task menu is implemented in index2.php. When the file is initially loaded it displays a selection box that is populated with all of the possible WISP tasks. Once a task is selected the file is reloaded and a form requesting the fields required for the selected task is displayed.

This task menu system allows several different tasks to be performed for a specific student without having to continually reselect the required information.

3.5.2 User Authentication

The first page the user sees when the WISP system is accessed is the login screen contained in index.php. Because of its name, this page automatically gets loaded

when the main directory is viewed with a web browser. This page gives the user three options. They can login, change their password, or have their password reset.

To login, the user enters their email address and password. If their email address is not found in the database, authentication fails and control is passed back to index.php with a variable set to indicate that the login attempt failed. If the email address exists in the database an encrypted version of the password entered is compared to the password stored in the record with that email address. The encryption is accomplished using the MySQL password function. If the passwords match, control is transferred to isp.php, otherwise control is once again transferred to index.php with an indication that the login process has failed.

Once the user is authenticated, a session variable is set using the cookie mechanism. This allows each page to verify that the user has successfully logged before granting them access to information.

Changing a password is handled by the files change_passwd.php and change_passwd2.php. In order for the user to change their password they must enter their email address, their current password, and their new password twice. If the user is authenticated and the two versions of the new password match, the user's password is changed

and an encrypted version of the new password is stored in the database.

The third choice the user has is to request a new password. This is useful if they have forgotten their password and need a new one. The files get_passwd.php and get_passwd2.php are used to accomplish this task. The user submits their email address and if it exists in the database, a new password is generated and emailed to that address. An encrypted version of the new password is then stored in the database.

The file authenticate.php contains functions used to authenticate users. The file is included by the programs concerned with logging in, changing passwords, or resetting passwords.

The code in valid_user.php uses PHP sessions to verify whether the current user has successfully logged in. If it is determined that the user has not logged in, control is transfered to isp.php and the user is asked to log in.

3.5.3 Data Entry and Reporting

Data entry includes creating course of studies, monthly evaluations, and semester report cards. Reporting consists of displaying these results as well as graduation plans and transcripts.

3.5.3.1 Course of Study.

Creating and displaying a course of study is handled by the files cos.php, cos2.php,

cos3.php, deleteclass.php, deleteclasscheck.php, and showcos.php.

The file cos.php displays the current course of study for a specific student, grade level, and semester. If the page is loaded with the variable $printfmt set to zero, the course of study will be displayed in a format that will allow editing, otherwise it will be loaded as a view-only page ready for printing.

When the page is loaded in edit mode, the user can add a class, edit a class, or delete a class. To add a class to the course of study, the user must select a subject from the list and click the "Add a Class" button, which will transfer control to cos2.php. To delete an existing class from the course of study, the user clicks the "Del" button to the left of the class they desire to delete. This action will transfer control to deleteclasscheck.php which will ask for confirmation of the delete action. If the user continues, the file deleteclass.php is called and the class is deleted, otherwise cos.php is loaded and the deletion process is skipped.

The file cos2.php is called from cos.php to add a new class to the course of study or to edit existing class information. If the variable $cosid has a non-zero value, the information associated with that value is read from the courseofstudy database table and presented in the form as information to be edited. If the value of $cosid is zero

each class in the course of study. Once this information is entered the user clicks "Submit" and control is passed to cos3.php.

The cos3.php page stores the information entered from eval2.php in the database and transfers control to showeval.php to display the resulting monthly evaluation in a format ready to be printed.

3.5.3.3 Semester Report Cards. Semester report cards are created and displayed using the reportcard.php file. This program uses variable $printfmt to determine whether to edit or display the report card information. If $printfmt is set to 1, the report card is displayed as a view-only page ready to be printed. If $printfmt is anything other than 1, the user is asked to assign or edit the grades and units for each class. Once the user clicks "Save Grades" control is passed to reportcard2.php.

The program reportcard2.php stores the grade and unit information in the database and transfers control back to reportcard.php with $printfmt set to 1 which causes the report card to be displayed. A report card can also be displayed by calling the file showreportcard.php with the variables $studentid, $gradelevel, and $semester set appropriately, and $printfmt set to 1. This method is used by the main task menu.

3.5.3.4 Graduation Plans. The file showgradplan.php creates a high school graduation plan from the course of

studies entered for a specific student. The plan shows every class that the student is planning to take and when they are planing to take it. The plan also displays the total number of units the student is planning to take in each subject area and the number required to graduate. The plan is displayed in a format that can be printed.

3.5.3.5 High School Transcripts. A preliminary high school transcript can be displayed by calling showtranscript.php with a valid $studentid. The program reads all of the attendance, grade, and unit information for the specific student and from the database and displays it.

3.5.4 Administrative Tasks

The ISP administrators have the ability to manage member information, print mailing labels, link subjects and classes, and make changes to the database as necessary.

The files that implement these tasks are located in the admin directory along with a file named .htaccess. The .htaccess file contains instructions that tell Apache to allow access to the admin directory only to authorized users. (See Setting up Administrative Security on page 44.)

The file index.php in this directory contains the main administration menu. (See Figure 12.)

3.5.4.1 Manage Member Information. The files addmember1.php and addmember2.php are used to add a new member to the database.

The updatemember1.php program displays a form that requests the name of the person to be updated. Once the administrator clicks the "Next" button, control is transferred to updatemember2.php which selects all of the members that meet the search criterion entered in updatemember1.php. If the number of matches is 1, control is transferred to addmember1.php with $memberid set to the id of the member matched. If more than one member is matched, a selection box is created with all of the matches and the administrator is prompted to select one. Once a selection is made and the "Edit" button is clicked, control is transfered to addmember1.php with $memberid set to the id of the member selected.

Member Information can be displayed using addressrpt.php or membershiplabels.php. The addressrpt.php program displays a form that requests the name of the person to be listed. It also contains check boxes which can be independently selected to specify whether phone, address, or email address should be included on the report. The query is set up to append a wild card to the end of the name being searched. This means that if a partial name is entered, multiple records may be displayed. It also means that if the fields are left blank, all of the members will be displayed. Once the "Submit" button is clicked, control is transferred to showmembers.php which retrieves the requested information from the database and displays it.

Another method of displaying member information is with the program membershiplabels.php. This program creates a file in PFD format that can be used to print mailing labels. The program uses a class from an open source project called PDF-PHP [13] to create the PDF file. PDF-PHP might be useful for creating a final version of the transcript.

3.5.4.2 Manage the Database. The file linksubjecttoclasses.php can be used by the addministrator to make changes to the subject_classes table. The program associates classes with subjects. When the program is first run it presents a list of subjects and asks the administrator to select one. Once the "Submit" button is clicked, the page is called again with the variable $subjectid set. This time a list of all of the classes is displayed with all of the classes currently linked to the subject selected. Additional classes can be added to those currently selected by holding down the control key while clicking on the class name. Classes can be unlinked in the same way. Once the "Submit" button is clicked, the page is called again for the third time. This time the variable $subjectid and $classid will both be set so the program stores the appropriate data in the subject_classes table and reports the number of classes linked to the selected subject.

CHAPTER FOUR

DATABASE DESIGN

4.1 Database Language and Tools

The database used in this project is MySQL. The decision to use MySQL was based on several factors. MySQL is open source, relatively easy to use, fast, robust, and integrates well with Apache and PHP.

In order to work efficiently with MySQL a tool such as phpMyAdmin should be used. phpMyAdmin is a server side program that allows database functions to be executed remotely from any computer with a web browser and valid authentication.

DBDesigner4 [2] was used to create the Entity Relationship Diagram (ERD) [12] shown in the next section. The program is normally used to design a database system and then create the structured query language (SQL) to create the tables in the database. It has a command to reverse engineer an existing database which proved to be extremely valuable in documenting the WISP database.

4.2 Entity Relationship Diagram

The entity relationship diagram in Figure 21 was produced from the existing WISP database using DBDesigner4.

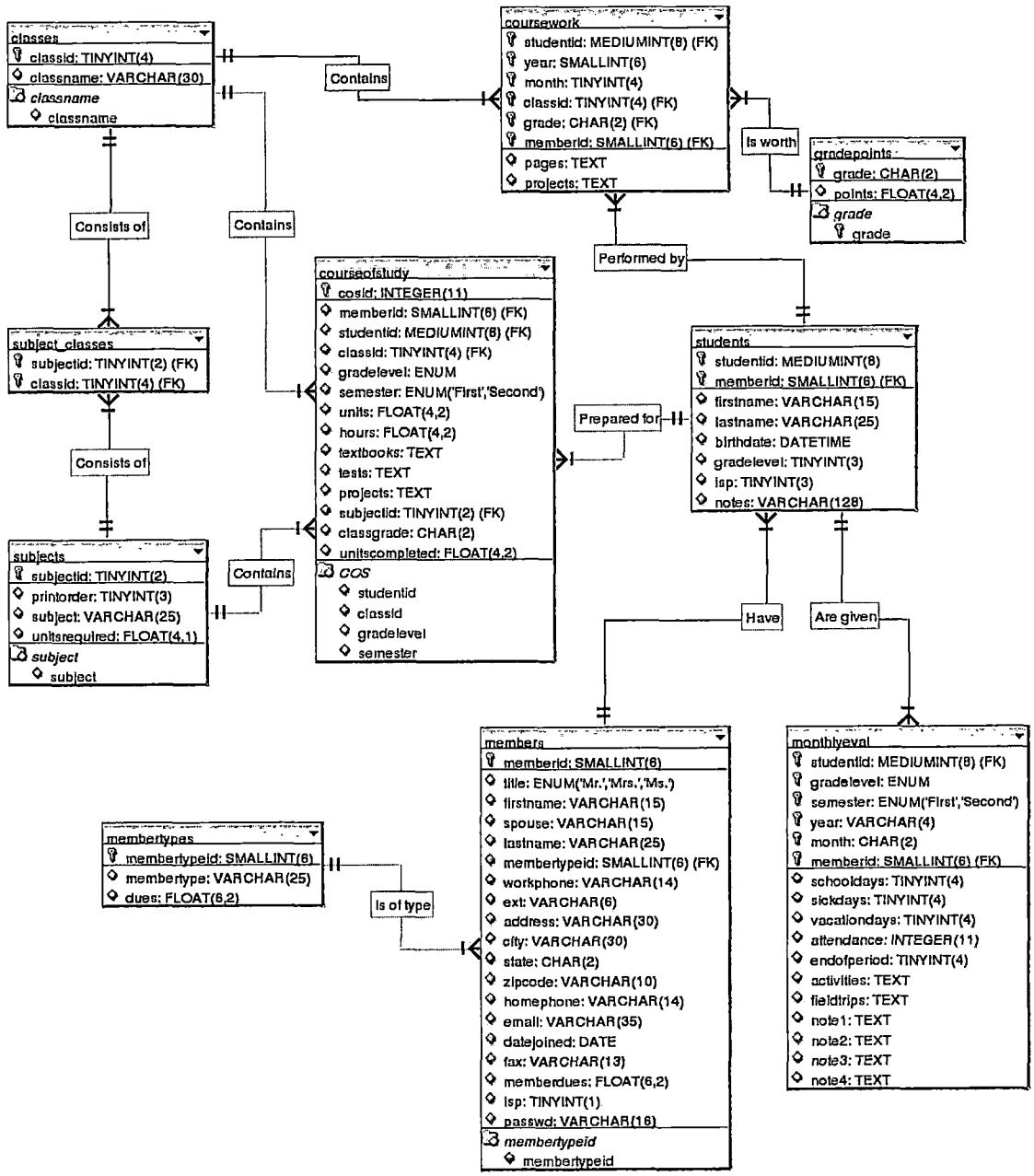


Figure 21. Entity Relationship Diagram

4.3 Database Schema

The database schema listed in Appendix B was generated from the WISP database using phpMyAdmin.

CHAPTER FIVE

INSTALLATION AND MAINTENANCE

5.1 System Installation

The process of installing the WISP system can be divided into these main tasks: loading the Linux operating system, testing Apache, MySQL, and PHP functionality, creating the WISP database and loading it with data, and loading the WISP program files. These tasks will be described in detail.

5.1.1 Loading the Linux Operating System

The Linux distribution of choice for this project is Mandrake Linux 10. This choice was originally made because Mandrake binaries are optimized for Pentium class processors while other popular distributions are optimized only for 386 class processors. Mandrake has a reputation for being easy to install, stable, and complete.

A current Mandrake distribution can be obtained in several ways. It can be purchased as a boxed set for approximately $80 from a computer retail outlet, purchased with a book or magazine for approximately $10 to $30 from a book store, or downloaded from the internet for approximately $0.

Once a Linux distribution is obtained, it can be installed by booting the system with the CD or DVD and following directions. During the installation process,

check the option to select individual packages and verify that Apache, MySQL, and PHP are selected to be loaded. Near the end of the installation there is an opportunity to specify the services that will be started automatically at run time. Verify that httpd and mysql are checked. If necessary, these services can be added to the default run levels manually using these commands:

```
chkconfig --add httpd
chkconfig --add mysql
```

If a service needs to be stopped or started use:

```
services httpd stop
services httpd start
```

5.1.2 Testing Software Functionality

To verify that Apache and MySQL are running, execute the command: "ps -xa" and look for httpd2 and mysqld. If either one of these are not running, try to start it using the services command shown above. If both services are running, test the installation by creating a file in your document root directory named php_info.php containing only this one line:

```
<? php_info(); ?>
```

Start a web browser and load the file by going to:

```
http://www.dikaios.org/php_info.php
```

If Apache and PHP are configured properly, a detailed information screen similar to Figure 22 will be displayed.

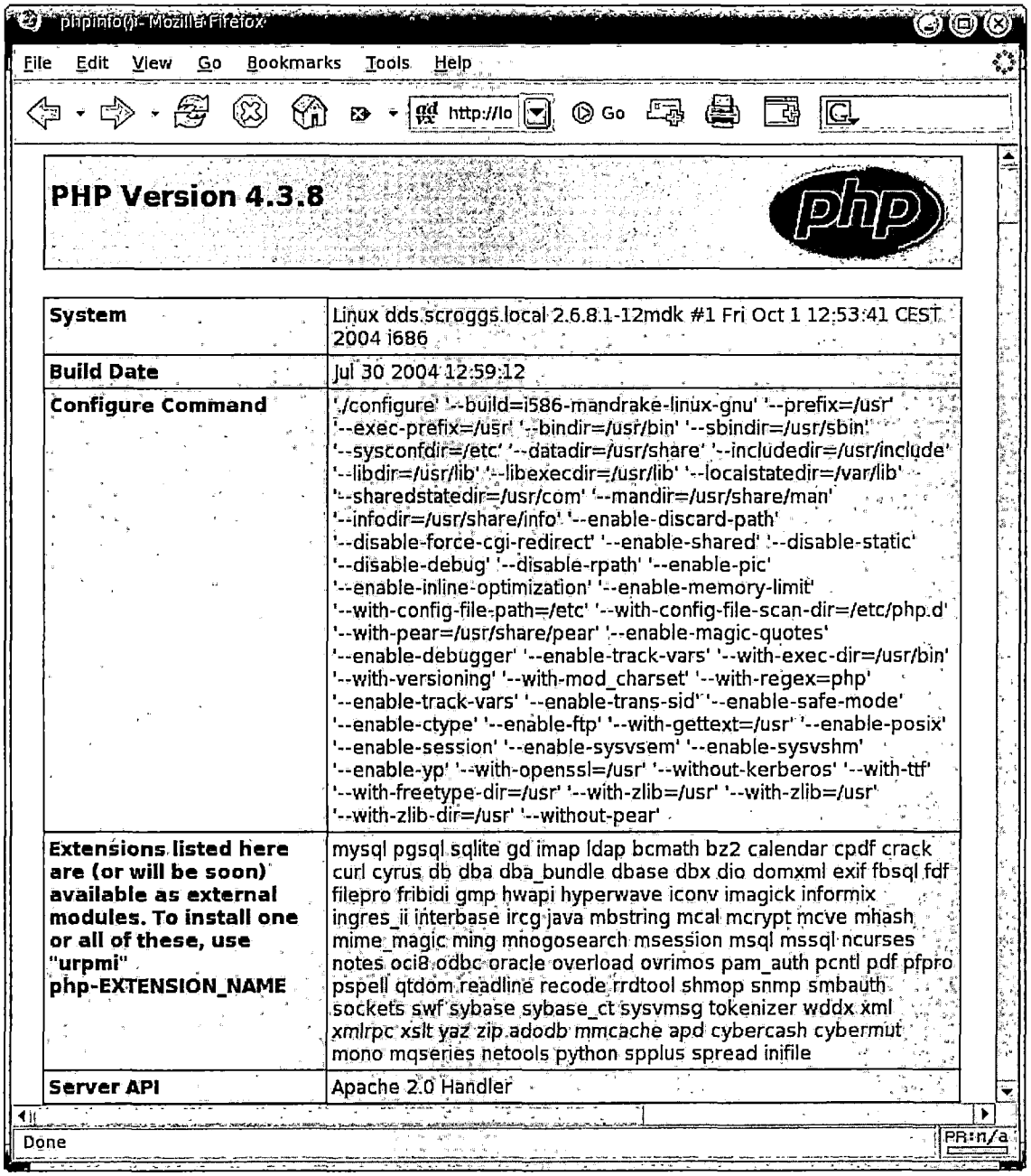


Figure 22. Output of the php_info Function

5.1.3 Creating the Database

The WISP database can be created from the schema file wisp.sql. Use the following commands in a shell:

```
mysql -u root -p < wisp.sql
```

The system will prompt for a password. Once the password is entered all of the commands in wisp.sql will be executed.

To create the WISP database with the data use the file wispanddata.sql instead of the file wisp.sql in the command above.

5.1.4 Loading the Program Files

This task involves copying the files from the WISP CD-ROM to the proper location on the server and changing the permissions on the files. After copying the files compare the long directory listing with the one in the file WISP_files.

5.1.5 Setting up Administrative Security

The WISP program uses a password system for regular users that does not need any special setup. For administrator security, WISP uses the authentication provided by Apache and htaccess. The htaccess method has been tested over a long period of time by many users and is known to be reliable. The process involves placing a file named .htaccess in the admin directory. This is the contents of the file:

```
AuthName "Dikaios information"
AuthType Basic
```

```
AuthUserFile    /home/httpd/.isp
<LIMIT GET POST>
     require valid-user
</LIMIT>
```

The AuthUserFile shows the location and file name of the file containing the usernames and passwords for the users who have access to the protected directory. This file is created with the following command:

/usr/sbin/htpasswd -c /home/httpd/.isp admin_user

where admin_user should be replaced with the desired user name of one of the administrators. The htpasswd command will prompt for a password twice. This will become the password for the newly created admin_user login.

To add additional administrative logins, use the same command but without the -c create option. The command should look like this:

/usr/sbin/htpasswd /home/httpd/.isp admin_user2

5.2 System Maintenance

The WISP system is currently virtually hosted on a server owned by Bell Enterprise in San Bernardino, California. System maintenance is composed of making complete nightly back-ups of the programs and the database. If the system were to crash or become corrupted, it could be rebuilt by following the instructions above and using the most recent back-up in the /backups directory on the

backup workstation. Using this backup strategy, it should not be possible to lose more than one days worth of data. If this potential loss is considered unacceptable, then database replication should be considered.

CHAPTER SIX

FUTURE ENHANCEMENTS

6.1 Immediate Enhancements

The WISP system will be used by approximately 100 home educators starting August 2005. There are several enhancements that should be made to the system before that date.

6.1.1 Create Plans from Existing Plans

One of the first additions that should be made to the system is to allow a course of study or complete graduation plan to be created by copying it from one previously entered. This should be very helpful because the second semester's course of study usually looks very similar to the first semester's and a graduation plan created for one child is probably a great starting point for creating one for a sibling.

This feature can be extended further by creating a few generic example entries that can be copied as a starting point for customization.

6.1.2 Archiving Final Versions

The ISP is required to keep certain records for all of the students in the ISP. Hard copies of the records submitted by the home educator are archived in fire proof boxes. The records can be used to document a student's educational history if it is ever questioned. The course

of study, graduation plan, monthly evaluation, and grade report are the records currently archived. It is important that the versions of these forms accurately reflect the latest records of each student.

An enhancement that would insure that the ISP always had the latest version of these forms would be to add three timestamp fields to each record; time of last change, time of last submission, and time of last archival. Once a form is completed and submitted, the time of last submission would be updated. Whenever the date of submission is more recent than the date of archival, the ISP administrator could be notified by email or through the use of a WISP report. Once the administrator knows that the form has been completed, it could be printed and archived which would update the archival timestamp.

If the home educator later changes any form the time of last change would be updated and indicate that the current version of the form has not been archived and must be resubmitted.

6.2 Not So Immediate Enhancements

There are also a few enhancements that would be nice but are not needed immediately.

6.2.1 Accounting

Dikaios currently keeps track of member payments and sends accounting statements to members who have outstanding

balances. In the future this process should be handled by the WISP system.

6.2.2 Broadcast Email

There are several groups of students, such as boys basketball, girls volleyball, student clubs, etc., that receive mail from Dikaios on a recurring basis. It would be very helpful if the WISP system would allow the ISP administrator or other designated individuals to send email to these groups. The system could also allow email to be sent to the entire membership. This would make it possible to send out a digital version of the monthly newsletter or to alert members of upcoming events.

CHAPTER SEVEN

CONCLUSION

7.1 Using Open Source

The open source software packages used to create this project (Linux, Apache, MySQL, and PHP) all functioned flawlessly. They are an extremely powerful combination, that should be seriously considered by anyone developing a web-based database application.

7.2 Project Conclusion

The WISP system as it is currently implemented allows home educators to create, store, edit, view, and print the forms and records associated with home schooling that are required by the Dikaios ISP. These forms include the student's course of study, monthly evaluations, semester grade reports, graduation plan, and transcripts.

The WISP project should be well received by the Dikaios home educators as a tool to simplify and improve the record keeping process.

APPENDIX A

SAMPLE OUTPUT

A.1 Course of Study

Course of Study for Jennifer Scroggins
Grade Level 9 - First Semester

Subject Area	Class	Textbooks	Tests	Papers/Projects
Math	Algebra I Units: 5.00 Hours: 5.00	BJU Algebra I	Weekly Chapter Tests	Daily Assignments
Elective	Art Units: 2.00 Hours: 3.00	Painting like the Masters	None.	Three paintings
Elective	Bible 9 Units: 2.50 Hours: 2.50	WWJD Today? by Helen Haidle Includes scripture study and application.	N/A	Daily assignments, sermons, church youth activities, outreach events, Bible Study at Co-op, etc.
Science	Biology Units: 5.00 Hours: 6.00	BJU Biology Text & Lab Workbook	Chapter Tests	Labs 2-3 times/week
English	English 9 Units: 5.00 Hours: 5.00	BJU English 9, Writing and Grammar BJU English Handbook Worldly Wise 3000, book5	Chapter Tests for both books	Daily assignments, one research project, letters, journal, writing contest.
Geography	Geography Units: 5.00 Hours: 5.00	Bob Jones Geography	T=5 Q=5	One 5 page paper
Elective	Home Econmics Units: 5.00 Hours: 5.00	Various First Aid Books	Weekly tests and Final Exam	Develop home management
Fine Arts	Music Units: 2.50 Hours: 3.00	Music Theory Book Piano lessons using various song books	T=10	Recital
Physical Education	PE Units: 5.00 Hours: 5.00		T=0 Q=0	Volleyball

A.2 Monthly Evaluation

Dikaios Christian Academy 2003-2004
Monthly Evaluation Form

Name: Jennifer Scroggins Grade Level: 9 Date: 09/2003

School Days: 22 Sick Days: 0 Vacation Days: 0 Total School Days: 55

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
☑	☑	☑	☑	☑	☐	☐	☑	☑	☑	☑	☑	☐	☐	☑	☑
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
☑	☑	☑	☐	☐	☑	☑	☑	☑	☑	☐	☐	☑	☑	☐	

Subject	Grade	Pages or Units	Projects & Comments
Algebra I	A	1-100	None
Art	B	Lots	Lots
Bible 9	A		Leads weekly devotions
Biology	A		Dissecting
English 9	B	100-200	Speech
Geography	A	Ch 12	Made a map
Home Econmics	A		
Music	B+		
PE	B		Volleyball
Science Lab	A		

Extra-curricular Activities

Activities:	
Fieldtrips:	
Note1:	
Note2:	
Note3:	
Note4:	

A.3 High School Graduation Plan

Dikaios Christian Academy

High School Graduation Plan for: Jennifer Scroggins

Subject Area	9th Grade		10th Grade		11th Grade		12th Grade		Total Units	
	1st Sem	2nd Sem	1st Sem	2nd Sem	1st Sem	2nd Sem	1st Sem	2nd Sem	Planned	Required
Math	Algebra I 5.00 units	Algebra I 5.00 units	Geometry 5.00 units	Geometry 5.00 units	Algebra II 5.00 units	Algebra II 5.00 units			30.0	20.0
English	English 9 5.00 units	English 9 5.00 units	English 10 5.00 units	English 10 5.00 units	English 11 5.00 units	English 11 5.00 units	English 12 5.00 units	English 12 5.00 units	40.0	30.0
Science	Biology 5.00 units Science Lab 0.00 units	Biology 5.00 units	Basic Science 5.00 units Science Lab 0.00 units	Basic Science 5.00 units Science Lab 0.00 units			Equine Science 5.00 units	Gardening 5.00 units	30.0	20.0
World History			World History 5.00 units	World History 5.00 units					10.0	10.0
U.S. History							U.S. History 5.00 units	U.S. History 5.00 units	10.0	10.0
Geography	Geography 5.00 units	Geography 5.00 units							10.0	5.0
U.S. Government					Government 5.00 units				5.0	5.0
Economics						Economics 5.00 units			5.0	5.0
Fine Arts	Music 2.50 units						Photography 5.00 units		7.5	10.0
Foreign Language			French I 5.00 units	French I 5.00 units	French II 5.00 units	French II 5.00 units			20.0	10.0
Health							Health 2.50 units	Health 2.50 units	5.0	5.0
Physical Education	PE 5.00 units	PE 5.00 units	PE 5.00 units	PE 5.00 units	PE 5.00 units		PE 5.00 units		30.0	20.0
Elective	Bible 9 2.50 units Home Econmics 5.00 units Art 2.00 units	Home Econmics 5.00 units Bible 9 2.50 units Music 5.00 units	Bible 10 2.50 units Music 2.50 units	Bible 10 2.50 units	Bible 11 2.50 units Tutoring 2.50 units	Bible 11 2.50 units Tutoring 2.50 units PSAT/SAT Prep 5.00 units	Other 5.00 units Bible 12 2.50 units	Bible 12 2.50 units Other 5.00 units	59.5	90.0
Note that any units taken in a specific subject area above the requirement may be counted as electives. This means that the number of elective units required do not have to be met as long as the total number of units taken meet the total number required.							**Grand Total:**		**262.0**	**240.0**

A.4 Report Card

Dikaios Christian Academy

Report Card for: Jennifer Scroggins Grade Level: 9 - First semester

Subject Area	Class	Hours	Units Attempted	Units Completed	Grade
Math	Algebra I	5.00	5.00	5.00	A-
Elective	Art	3.00	2.00	2.00	A
Elective	Bible 9	2.50	2.50	2.50	A
Science	Biology	6.00	5.00	5.00	A-
English	English 9	5.00	5.00	5.00	A
Geography	Geography	5.00	5.00	5.00	B+
Elective	Home Econmics	5.00	5.00	5.00	A
Fine Arts	Music	3.00	2.50	2.50	A-
Physical Education	PE	5.00	5.00	5.00	A
Science	Science Lab	2.00	0.00	0.00	A-
	Totals:	41.50	37.00	37.00	3.80
EXTRA-CURRICULAR ACTIVITIES					
Family trip to Nebraska and Kansas. Spent 5 days at the family cattle ranch.					

Teacher's Signature: _____ Date: _____

ISP Admin's Signature: _____ Date: _____

A.5 High School Preliminary Transcript

Dikaios Christian Academy

Transcript for: _Jennifer Scroggins_

Total Attendance
Sick Days: _2_ Vacation Days: _1_ Days in School: _179_

Credit Summary

Subject Area	Required	Completed	Deficient
Math	20.0	25.0	0.0
English	30.0	25.0	5.0
Science	20.0	20.0	0.0
World History	10.0	10.0	0.0
U.S. History	10.0	0.0	10.0
Geography	5.0	10.0	0.0
U.S. Government	5.0	5.0	0.0
Economics	5.0	0.0	5.0
Fine Arts	10.0	2.5	7.5
Foreign Language	10.0	15.0	0.0
Health	5.0	0.0	5.0
Physical Education	20.0	25.0	0.0
Elective	90.0	34.5	55.5
Total Credits	240.0	172.0	68.0

Grade Summary

Subject Area	Class	Hours	Units Attempted	Units Completed	Grade
Math	Algebra I	5.00	5.00	5.00	A-
English	English 9	5.00	5.00	5.00	A
Science	Biology	6.00	5.00	5.00	A-
Elective	Bible 9	2.50	2.50	2.50	A
Elective	Home Econmics	5.00	5.00	5.00	A
Elective	Art	3.00	2.00	2.00	A

Fine Arts	Music	3.00	2.50	2.50	A-
Science	Science Lab	2.00	0.00	0.00	A-
Geography	Geography	5.00	5.00	5.00	B+
Physical Education	PE	5.00	5.00	5.00	A
Math	Algebra I	5.00	5.00	5.00	A-
English	English 9	5.00	5.00	5.00	A
Science	Biology	5.00	5.00	5.00	A-
Geography	Geography	5.00	5.00	5.00	B+
Physical Education	PE	5.00	5.00	5.00	A
Elective	Home Econmics	5.00	5.00	5.00	A
Elective	Bible 9	2.50	2.50	2.50	A
Elective	Music	5.00	5.00	5.00	A
Elective	Bible 10	3.00	2.50	2.50	A
Math	Geometry	5.00	5.00	5.00	A-
English	English 10	5.00	5.00	5.00	A
Foreign Language	French I	5.00	5.00	5.00	B+
Science	Basic Science	5.00	5.00	5.00	A
World History	World History	5.00	5.00	5.00	A
Physical Education	PE	5.00	5.00	5.00	A
Elective	Music	3.00	2.50	2.50	A
Science	Science Lab	2.00	0.00	0.00	A-
Elective	Bible 10	3.00	2.50	2.50	A
Science	Basic Science	5.00	5.00	5.00	A-
Science	Science Lab	3.00	0.00	0.00	B+
Math	Geometry	5.00	5.00	5.00	B
English	English 10	10.00	5.00	5.00	A-
World History	World History	5.00	5.00	5.00	A-
Foreign Language	French I	5.00	5.00	5.00	A
Physical Education	PE	5.00	5.00	5.00	A-
Math	Algebra II	5.00	5.00	5.00	A-
U.S. Government	Government	5.00	5.00	5.00	B+
English	English 11	5.00	5.00	5.00	A
Physical Education	PE	5.00	5.00	5.00	A

Foreign Language	French II	5.00	5.00	5.00	B+
Elective	Bible 11	2.50	2.50	2.50	A
Elective	Tutoring	2.50	2.50	2.50	A
English	English 11	5.00	5.00	0.00	
Economics	Economics	5.00	5.00	0.00	
Math	Algebra II	5.00	5.00	0.00	
Foreign Language	French II	5.00	5.00	0.00	
Elective	Bible 11	2.50	2.50	0.00	
Elective	Tutoring	2.50	2.50	0.00	
Elective	PSAT/SAT Prep	5.00	5.00	0.00	
Elective	Other	5.00	5.00		
English	English 12	5.00	5.00		
U.S. History	U.S. History	5.00	5.00		
Physical Education	PE	5.00	5.00		
Health	Health	2.50	2.50		
Elective	Bible 12	2.50	2.50		
Fine Arts	Photography	5.00	5.00		
Science	Equine Science	5.00	5.00		
English	English 12	5.00	5.00		
U.S. History	U.S. History	5.00	5.00		
Health	Health	2.50	2.50		
Elective	Bible 12	2.50	2.50		
Science	Gardening	5.00	5.00		
Elective	Other	5.00	5.00		
			Totals:	172.00	3.78
EXTRA-CURRICULAR ACTIVITIES					
Family trip to Nebraska and Kansas. Spent 5 days at the family cattle ranch.					

Teacher's Signature: _____ Date: _____

ISP Admin's Signature: _____ Date: _____

APPENDIX B

DATABASE SCHEMA

```
-- --------------------------------------------------------
--
-- phpMyAdmin SQL Dump
-- version 2.6.2-rc1
-- http://www.phpmyadmin.net
--
-- Host: localhost
-- Generation Time: Apr 28, 2005 at 12:32 AM
-- Server version: 4.0.20
-- PHP Version: 4.3.8
--
-- WISP Schema
--
--
-- Database: `dikaios`
--

CREATE DATABASE `dikaios`;
USE dikaios;


-- --------------------------------------------------------

--
-- Table structure for table `classes`
--

CREATE TABLE IF NOT EXISTS `classes` (
  `classid` tinyint(4) NOT NULL auto_increment,
  `classname` varchar(30) NOT NULL default '',
  PRIMARY KEY  (`classid`),
  KEY `classname` (`classname`)
) TYPE=MyISAM AUTO_INCREMENT=110 ;


-- --------------------------------------------------------

--
-- Table structure for table `courseofstudy`
--

CREATE TABLE IF NOT EXISTS `courseofstudy` (
  `cosid` int(11) NOT NULL auto_increment,
  `studentid` mediumint(8) unsigned NOT NULL default '0',
  `classid` tinyint(4) unsigned NOT NULL default '0',
  `gradelevel` enum
('K','1','2','3','4','5','6','7','8','9','10','11','12') default NULL,
  `semester` enum('First','Second') NOT NULL default 'First',
  `units` float(4,2) unsigned NOT NULL default '0.00',
  `hours` float(4,2) unsigned NOT NULL default '0.00',
  `textbooks` text NOT NULL,
  `tests` text NOT NULL,
  `projects` text NOT NULL,
  `subjectid` tinyint(2) unsigned NOT NULL default '0',
  `classgrade` char(2) NOT NULL default '',
  `unitscompleted` float(4,2) unsigned default NULL,
  PRIMARY KEY  (`cosid`),
  UNIQUE KEY `COS` (`studentid`,`classid`,`gradelevel`,`semester`)
) TYPE=MyISAM AUTO_INCREMENT=193 ;
```

```
-- --------------------------------------------------------

--
-- Table structure for table `coursework`
--

CREATE TABLE IF NOT EXISTS `coursework` (
  `studentid` smallint(6) NOT NULL default '0',
  `year` smallint(6) NOT NULL default '0',
  `month` tinyint(4) NOT NULL default '0',
  `classid` smallint(6) NOT NULL default '0',
  `grade` char(2) NOT NULL default '',
  `pages` text NOT NULL,
  `projects` text NOT NULL,
  PRIMARY KEY  (`studentid`,`year`,`month`,`classid`)
) TYPE=MyISAM;


-- --------------------------------------------------------

--
-- Table structure for table `gradepoints`
--

CREATE TABLE IF NOT EXISTS `gradepoints` (
  `grade` char(2) NOT NULL default '',
  `points` float(4,2) NOT NULL default '0.00',
  UNIQUE KEY `grade` (`grade`)
) TYPE=MyISAM;


-- --------------------------------------------------------

--
-- Table structure for table `members`
--

CREATE TABLE IF NOT EXISTS `members` (
  `memberid` smallint(6) NOT NULL auto_increment,
  `title` enum('Mr.','Mrs.','Ms.') NOT NULL default 'Mr.',
  `firstname` varchar(15) NOT NULL default '',
  `spouse` varchar(15) NOT NULL default '',
  `lastname` varchar(25) NOT NULL default '',
  `membertypeid` smallint(6) NOT NULL default '0',
  `workphone` varchar(14) NOT NULL default '',
  `ext` varchar(6) NOT NULL default '',
  `address` varchar(30) NOT NULL default '',
  `city` varchar(30) NOT NULL default '',
  `state` char(2) NOT NULL default '',
  `zipcode` varchar(10) NOT NULL default '',
  `homephone` varchar(14) NOT NULL default '',
  `email` varchar(35) NOT NULL default '',
  `datejoined` date NOT NULL default '0000-00-00',
  `fax` varchar(13) NOT NULL default '',
  `memberdues` float(6,2) NOT NULL default '0.00',
  `isp` tinyint(1) NOT NULL default '0',
  `passwd` varchar(16) binary default NULL,
  PRIMARY KEY  (`memberid`),
  KEY `membertypeid` (`membertypeid`)
) TYPE=MyISAM AUTO_INCREMENT=270 ;
```

```
-- --------------------------------------------------------

--
-- Table structure for table `membertypes`
--

CREATE TABLE IF NOT EXISTS `membertypes` (
  `membertypeid` smallint(6) NOT NULL auto_increment,
  `membertype` varchar(25) NOT NULL default '',
  `dues` float(6,2) NOT NULL default '0.00',
  PRIMARY KEY  (`membertypeid`)
) TYPE=MyISAM AUTO_INCREMENT=9 ;


-- --------------------------------------------------------

--
-- Table structure for table `monthlyeval`
--

CREATE TABLE IF NOT EXISTS `monthlyeval` (
  `studentid` smallint(5) unsigned NOT NULL default '0',
  `gradelevel` enum
('K','1','2','3','4','5','6','7','8','9','10','11','12') NOT NULL
default 'K',
  `semester` enum('First','Second') NOT NULL default 'First',
  `year` varchar(4) NOT NULL default '0',
  `month` char(2) NOT NULL default '0',
  `schooldays` tinyint(4) NOT NULL default '0',
  `sickdays` tinyint(4) NOT NULL default '0',
  `vacationdays` tinyint(4) NOT NULL default '0',
  `attendance` int(11) NOT NULL default '0',
  `endofperiod` tinyint(4) NOT NULL default '0',
  `activities` text NOT NULL,
  `fieldtrips` text NOT NULL,
  `note1` text NOT NULL,
  `note2` text NOT NULL,
  `note3` text NOT NULL,
  `note4` text NOT NULL,
  PRIMARY KEY  (`studentid`,`year`,`month`,`gradelevel`,`semester`)
) TYPE=MyISAM;


-- --------------------------------------------------------

--
-- Table structure for table `students`
--

CREATE TABLE IF NOT EXISTS `students` (
  `studentid` mediumint(8) unsigned NOT NULL auto_increment,
  `memberid` mediumint(9) NOT NULL default '0',
  `firstname` varchar(15) default NULL,
  `lastname` varchar(25) NOT NULL default '',
  `birthdate` datetime default NULL,
  `gradelevel` tinyint(3) unsigned default NULL,
  `isp` tinyint(3) unsigned NOT NULL default '0',
  `notes` varchar(128) default NULL,
  PRIMARY KEY  (`studentid`)
) TYPE=MyISAM AUTO_INCREMENT=601 ;
```

```
-- --------------------------------------------------------

--
-- Table structure for table `subject_classes`
--

CREATE TABLE IF NOT EXISTS `subject_classes` (
  `subjectid` tinyint(2) unsigned NOT NULL default '0',
  `classid` tinyint(3) unsigned NOT NULL default '0',
  PRIMARY KEY  (`subjectid`,`classid`)
) TYPE=MyISAM COMMENT='Links classes to subject areas';


-- --------------------------------------------------------

--
-- Table structure for table `subjects`
--

CREATE TABLE IF NOT EXISTS `subjects` (
  `subjectid` tinyint(2) unsigned NOT NULL auto_increment,
  `printorder` tinyint(3) unsigned NOT NULL default '0',
  `subject` varchar(25) NOT NULL default '',
  `unitsrequired` float(4,1) NOT NULL default '0.0',
  PRIMARY KEY  (`subjectid`),
  KEY `subject` (`subject`)
) TYPE=MyISAM PACK_KEYS=1 COMMENT='Subject Areas' AUTO_INCREMENT=20 ;
```

APPENDIX C

SOURCE CODE

```
<!-------------------------------------------------------
- Filename:  admin/addmember1.php
- Written by: Darryl Scroggins
- Last Modified : 2005-05-30 23:58
-------------------------------------------------------->
<html>
<head>
   <? include("../header.inc"); ?>
</head>

<body>

<?
require_once('../dbconnect.php');
$connect = dbconnect();

// If $memberid is set then update the member info otherwise add a new member
// echo "Memberid: ".$memberid."<br>";

if ($memberid>0) {
   $mode="Update";

   $q="SELECT * FROM members WHERE memberid=\"$memberid\"";

   $result = mysql_query($q) or die(mysql_error());
   $ncols=mysql_num_fields($result);
   if ($ncols==0) {
       die ("Member $memberid does not exist!");
   }

   $r = mysql_fetch_assoc($result);
   $title = $r["title"];
   $firstname = $r["firstname"];
   $spouse = $r["spouse"];
   $lastname = $r["lastname"];
   $curmembertypeid = $r["membertypeid"];
   $workphone = $r["workphone"];
   $ext = $r["ext"];
   $address = $r["address"];
   $city = $r["city"];
   $state = $r["state"];
   $zipcode = $r["zipcode"];
   $homephone = $r["homephone"];
   $email = $r["email"];
   $datejoined = $r["datejoined"];
   $fax = $r["fax"];
   $memberdues = $r["memberdues"];
   $isp = $r["isp"];
   $banknumber = $r["banknumber"];
} else {
   $mode="Add"; // Add a new member
}

//Make Membertype dropdown
$q="SELECT * FROM membertypes";
$result = mysql_query($q) or die(mysql_error());
$membertypes="<select size=\"1\" name=\"membertypeid\">";
while ($r = mysql_fetch_assoc($result)) {
   $membertype=trim($r['membertype']);
   if ($membertype<>"") {
      $membertypeid=$r['membertypeid'];
      $s="";
      if ($membertypeid==$curmembertypeid) { $s=" selected"; }
      $dues=$r['dues'];
      $membertypes.="<option value=$membertypeid$s>$membertype ($dues)</option>";
   }
}
$membertypes.="</select>";


//Make ISP options radio buttons
$yesno="<input type=\"radio\" name=\"isp\" value=\"1\" ".
    (($isp=="1")? " checked":"") ."> Yes ";
$yesno.="<input type=\"radio\" name=\"isp\" value=\"0\" ".
    (($isp=="0")? " checked":"") ."> No";
```

```
//Make ISP titles a dropdown from the enumerated types
$q="DESCRIBE members title";
$result = mysql_query($q) or die(mysql_error());
$r = mysql_fetch_assoc($result) or die(mysql_error());
$enum_string= $r['Type'];
$array_string=str_replace('enum', 'array', $enum_string);
eval("\$membertitles = $array_string;");
$titles="<select size=\"1\" name=\"title\">";
foreach($membertitles as $title_choice) {
   $s=""; //Default to not selected
   if ($title_choice==$title) { $s=" selected"; }
   $titles .= "<option value=\"$title_choice\"$s>$title_choice</option>";
}

$titles.="</select>";

echo "<center><h1>$mode Member Information</h1>";
?>

<form action="addmember2.php">
<input type="hidden" name="memberid" value="<? echo $memberid; ?>">
<center>
<table>
<tr><td align="right">
Title
</td><td>
<? echo $titles; ?></td></tr>
<tr><td align="right">
Firstname
</td><td>
<input type="text" name="firstname" value="<? echo $firstname; ?>"></td></tr>
<tr><td align="right">
Spouse
</td><td>
<input type="text" name="spouse" value="<? echo $spouse; ?>"></td></tr>
<tr><td align="right">
Membertype
</td><td>
<? echo $membertypes; ?></td></tr>
<tr><td align="right">
Work Phone
</td><td>
<input type="text" name="workphone" value="<? echo $workphone; ?>">
 Ext <input type="text" name="ext" size="6" value="<? echo $ext; ?>">
</td></tr>
<tr><td align="right">
Lastname
</td><td>
<input type="text" name="lastname" value="<? echo $lastname; ?>"></td></tr>
<tr><td align="right">
Street
</td><td>
<input type="text" name="address" value="<? echo $address; ?>"></td></tr>
<tr><td align="right">
City
</td><td>
<input type="text" name="city" value="<? echo $city; ?>"></td></tr>
<tr><td align="right">
State
</td><td>
<input type="text" name="state" value="<? echo $state; ?>"></td></tr>
<tr><td align="right">
Zipcode
</td><td>
<input type="text" name="zipcode" value="<? echo $zipcode; ?>"></td></tr>
<tr><td align="right">
Home Phone
</td><td>
<input type="text" name="homephone" value="<? echo $homephone; ?>"></td></tr>
<tr><td align="right">
E-mail
</td><td>
<input type="text" name="email" value="<? echo $email; ?>"></td></tr>
<tr><td align="right">
Date Joined
</td><td>
<input type="text" name="datejoined" value="<? echo $datejoined; ?>">
```

```
</td></tr>
<tr><td align="right">
FAX Number
</td><td>
<input type="text" name="fax" value="<? echo $fax; ?>">
</td></tr>
<tr><td align="right">
Member Dues
</td><td>
<input type="text" name="memberdues" value="<? echo $memberdues; ?>"></td></tr>
<tr><td align="right">
ISP Member?
</td><td>
<? echo $yesno; ?></td></tr>
<tr><td colspan="2" align="right">
<input type="submit" value="<? echo $mode; ?>"></td></tr>
</table>
</center>
</form>
</body>
</html>


<!-------------------------------------------------------
- Filename:  admin/addmember2.php
- Written by: Darryl Scroggins
- Last Modified : 2005-05-30 23:59
-------------------------------------------------------->
<?
require_once('../dbconnect.php');
$connect = dbconnect();

if ($memberid<=0) {
   die ("Invalid memberid: $memberid");
} else {
   $query= "UPDATE members SET title=\"$title\", firstname=\"$firstname\", ".
      "spouse=\"$spouse\", lastname=\"$lastname\", membertypeid=\"$membertypeid\", ".
      "workphone=\"$workphone\", ext=\"$ext\", address=\"$address\", ".
      "city=\"$city\", state=\"$state\", zipcode=\"$zipcode\", ".
      "homephone=\"$homephone\", email=\"$email\", datejoined=\"$datejoined\", ".
      "fax=\"$fax\", memberdues=\"$memberdues\", isp=\"$isp\" ".
      "WHERE memberid=\"$memberid\"";

   $result = mysql_query($query) or die(mysql_error());
   // header("Location: addmember1.php?memberid=$memberid");
   header("Location: index.php");
}


<!-------------------------------------------------------
- Filename:  admin/addressrpt.php
- Written by: Darryl Scroggins
- Last Modified : 2005-05-31 00:00
-------------------------------------------------------->
<html>
<head>
   <? include("../header.inc"); ?>
</head>

<body>
<br>
<center>
<h1>Member Information</h1>

<form method="POST" action=showmembers.php>

<table class="border">
<tr><td colspan="2">
Please enter the Lastname and optionally the Firstname of the member
that you would like to display.<br>
Leave both fields blank to see a list of all members<br><br>
</td>

<tr>
<td align="left" width="50%">
<table border=0>
<tr><td align="right" width="50%">
```

```
Lastname:
</td><td>
<input type="text" size="20" name="lastname" value="">
</td></tr>
<tr><td align="right">
Firstname:
</td><td>
<input type="text" size="10" name="firstname" value=""><br>
</td></tr>
<tr><td>
 
</td><td>
<input type="submit" value="Next">
</td></tr>
</table>
</td>

<td align="left" width="50%">
<!-- Start the table that holds the options checkboxes -->
<table>
<tr><td colspan="2">Check Information to Include</td></tr>
<tr>
<td align="right">
<input type="checkbox" name="address" value="1"></td>
<td>Address</td>
</tr>
<tr>
<td align="right">
<input type="checkbox" name="phone" value="1"></td>
<td>Phone</td>
</tr>
<tr>
<td align="right">
<input type="checkbox" name="email" value="1"></td>
<td>Email</td>
</tr>
</table>
</td>
</tr>

</table>
</form>
</center>
</body>
</html>


<!-------------------------------------------------------
- Filename:  admin/index.php
- Written by: Darryl Scroggins
- Last Modified : 2005-05-31 00:02
-------------------------------------------------------->
<html>
<head>
    <? include("../header.inc"); ?>
</head>

<body>
<center>
<h1>Welcome to Dikaios</h1>
<br>
<br>
<table class="border">
   <tr><td><ul>
      <li><a href="addmember1.php">Add a New Member</a><br><br>
      <li><a href="updatemember1.php">Update a Member's Information</a><br><br>
      <li><a href="addressrpt.php">View Address/Phone Information</a><br><br>
      <li><a href="membershiplabels.php">Create Membership Labels</a><br><br>
      <li><a href="linksubjecttoclasses.php">Link Subject to Classes</a><br><br>
      <li><a href="/phpMyAdmin/index.php">Manually Edit Database Tables</a><br><br>
      <li><a href="../index.php">Return to Main Dikaios Home Page</a>
   </ul></td></tr>
</table>
</center>

</body>
</html>
```

```
<!-------------------------------------------------------
- Filename:  admin/linksubjecttoclasses.php
- Written by: Darryl Scroggins
- Last Modified : 2005-05-31 00:04
-------------------------------------------------------->
<html>
<head>
    <? include("../header.inc"); ?>
</head>

<body>
<br><br><center>
<form method=\"POST\" action="linksubjecttoclasses.php">

<?
require_once('../dbconnect.php');
$connect = dbconnect();

if (!$subjectid) {
   $subjects = "";
   $query = "SELECT * FROM subjects";
   $result = mysql_query($query) or die("Bad query: $query");
   $size=mysql_num_rows($result);
   while ($r = mysql_fetch_array($result)){
      $subjectid=$r["subjectid"];
      $subject=$r["subject"];
      $subjects .= "<option value=\"$subjectid\">$subject</option>";
   }
   echo "Select a subject and press submit<br>";
   echo "<select name=\"subjectid\" size=\"$size\">";
   echo $subjects;
   echo "</select>";
} elseif (!$classid) {
   $query = "SELECT classid, classname FROM classes ORDER BY classname";
   $result = mysql_query($query) or die("Bad query: $query");
   $courses = "";
   while ($r = mysql_fetch_array($result)){
       $classid = $r["classid"];
       $classname = $r["classname"];
       $s=""; //default to NOT SELECTED
       $q2="SELECT classid FROM subject_classes WHERE subjectid=".
           "\"$subjectid\" AND classid=\"$classid\"";
       $r2 = mysql_query($q2) or die("Bad query: $q2");
       if (mysql_num_rows($r2)) { $s=" SELECTED"; }
       $courses .= "<option value=\"$classid\"$s>$classname</option>";
   }
   echo "Select all of the classes to be linked to the subject ".
       "and press link.<br> Use the Ctrl key while clicking classes ".
       "to select multiple classes.<br>";
   echo "<select name=\"classid[]\" size=\"15\" multiple>";
   echo $courses;
   echo "</select>";
   echo "<input type=\"hidden\" name=\"subjectid\" value=\"$subjectid\">";
} else {
   $query = "DELETE FROM subject_classes WHERE subjectid=\"$subjectid\"";
   $result = mysql_query($query);
   $num_selected = sizeof($classid);
   echo "Added ".$num_selected." classes to subject id: ".$subjectid."<br>";
   for ($i=0; $i<$num_selected; $i++) {
      $query="INSERT INTO subject_classes (subjectid, classid) ".
         "VALUES(\"$subjectid\", \"$classid[$i]\")";
      $result = mysql_query($query);
   }
}
?>

<br>
<input type="submit" value="Submit">
</form>

<br>
<a href="index.php">Back to Menu</a>
</center>
</body>
</html>
```

```
<!-------------------------------------------------------
- Filename:  admin/showmembers.php
- Written by: Darryl Scroggins
- Last Modified : 2005-05-31 00:10
-------------------------------------------------------->
<html>
<head>
   <? include("../header.inc"); ?>
</head>

<body>
<center>
<h1>Member Information<h1>

<?
require_once('../dbconnect.php');
$connect = dbconnect();

$ln=trim($lastname)."%";
$fn=trim($firstname)."%";

$psql="";
$asql="";
$esql="";

if ($address) { $asql=",address,city,state,zipcode"; }
if ($phone) { $psql=",homephone"; }
if ($email) { $esql=",email"; }

$sql="select lastname,firstname" .$psql. $asql.$esql." from members where ".
   "lastname LIKE \"$ln\"";

if ($fn!="%") {
   $sql.= " and firstname LIKE \"$fn\"";
}

$sql.= " order by lastname";

$result = mysql_db_query("dikaios",$sql) or
   die("Bad query: $sql<br>".mysql_errno() . ": " . mysql_error());

$n = mysql_num_rows($result);
if ($n==0) {
   echo "No members with lastname $fn and firstname $fn";
} else {
   echo "<table border=1><th>Firstname<th>Lastname";
   if ($address) echo "<th>Address";
   if ($phone) echo "<th>Phone";
   if ($email) echo "<th>email";

   $phonecol="";
   while ($r = mysql_fetch_array($result)){
      $firstname=$r["firstname"];
      if ($firstname=="") { $firstname=" "; }
      $lastname=$r["lastname"];
      if ($lastname=="") { $lastname=" "; }
      if ($address) {
          $addresscol="<td align='top'>".
          $r["address"].
          "<br>".
          $r[city].
          ", ".
          $r["state"].
          " ".
          $r["zipcode"].
          "</td>";
       }
       if ($phone) {
          $phonecol=$r["homephone"];
          if ($phonecol=="") { $phonecol=" "; }
          $phonecol="<td valign='top'>".$phonecol."</td>";
       }
       if ($email) {
          $emailcol=$r["email"];
          if ($emailcol=="") { $emailcol=" "; }
          $emailcol="<td valign='top'>".$emailcol."</td>";
       }
```

```
        echo "<tr>".
           "<td valign='top'>" . $firstname . "</td>".
           "<td valign='top'>" . $lastname  . "</td>".
           $addresscol .
           $phonecol .
           $emailcol .
           "</tr>\n";
    }
}
?>

</table>
</body>
</html>


<!-------------------------------------------------------
- Filename:  admin/updatemember1.php
- Written by: Darryl Scroggins
- Last Modified : 2005-05-31 00:10
-------------------------------------------------------->
<html>
<head>
   <? include("../header.inc"); ?>
</head>

<body>

<br>
<center>

<h1>Member Information</h1>

<form method="POST" action=updatemember2.php>
<table class="border">
<tr><td colspan=2>
Please enter the Lastname and optionally the Firstname of the
member that you would like to display.<br>
Leave both fields blank to see a list of all members
<br>
<br>

</td></tr>
<tr><td align="right">
Lastname:
</td><td>
<input type="text" size="20" name="lastname" value="">
</td></tr>
<tr><td align="right">
Firstname:
</td><td>
<input type="text" size="10" name="firstname" value="">
<br>
</td></tr>
<tr><td>
 
</td><td>
<input type="submit" value="Next">
</td>
</tr>
</table>

</form>

</center>
</body>
</html>


<!-------------------------------------------------------
- Filename:  admin/updatemember2.php
- Written by: Darryl Scroggins
- Last Modified : 2005-05-31 00:13
-------------------------------------------------------->
<?
require_once('../dbconnect.php');
```

```
$connect = dbconnect();

$ncols=0;
$ln=trim($lastname)."%";
$fn=trim($firstname)."%";

$sql="select memberid,lastname,firstname from members where ".
   "lastname LIKE \"$ln\"";

if ($fn!="%") {
   $sql.= " and firstname LIKE \"$fn\"";
}

$sql.= " order by lastname";

$result = mysql_db_query("dikaios",$sql) or
   die("Bad query: $sql<br>".mysql_errno() . ": " . mysql_error());

$ncols = mysql_num_fields($result);
$n = mysql_num_rows($result);
if ($n==0) {
   die("No members with lastname $lastname and firstname $firstname <br>");
}

if ($n==1) {
   $r = mysql_fetch_array($result);
   $memberid=$r["memberid"];
   header("Location: addmember1.php?memberid=$memberid");

} else {
   echo "<html>".
        "<head>";
   include("../header.inc");
   echo "</head>".
        "<body>".
        "<br><br><center>".
        "<h2>Please select a member to update and click \"Edit\".</h2>";

   echo "<form method=\"post\" action=\"addmember1.php\">";
   // Create select drop down
   $dropdown="<select size=\"1\" name=\"memberid\">";
   while ($r = mysql_fetch_array($result)){
        $firstname=$r["firstname"];
        $lastname=$r["lastname"];
        $memberid=$r["memberid"];
        $name= $lastname. ", ".$firstname;
        $dropdown.="<option value=$memberid>$name</option>";
   }
   $dropdown.="</select>";
   echo $dropdown;
   echo "<input type=\"submit\" value=\"Edit\">".
        "</form> </body> </html>";
}
?>


<!-------------------------------------------------------
- Filename:  authenticate.php
- Functions in this file were adapted from chapter 24 of [8]
- Last Modified : 2005-05-31 00:14
-------------------------------------------------------->
<?
require_once('dbconnect.php');

function register($email) {
  $connect = dbconnect();
  // Check if valid email
  $result = mysql_query("select memberid from members where email='$email'");
  if (!$result)
     return 'Could not execute query';
  if (mysql_num_rows($result)==0)
     return 'Could not find member with that email address';

   // Create password put it in database and email it
   $password=reset_password($email);
   if (!$password)
        return 'Could not create a password.';
```

```
   $result=notify_password($email, $password);
   if (!$result)
       return 'Could not register you in database.';

  return "success";
}

function login($email, $password) {
// check if email and password are in the database
// if they are return memberid otherwise return false
   $connect = dbconnect();
   $result = mysql_query("select memberid from members where email='$email'
                          and passwd = password('$password')");
   if (!$result) return false;
   if (mysql_num_rows($result)>0) {
      $r = mysql_fetch_assoc($result);
      $memberid=$r["memberid"];
      return $memberid;
   } else {
      return false;
   }
}

function check_valid_user() {
  if (isset($_SESSION['valid_user'])) {
      echo 'You are currently logged in as '.$_SESSION['valid_user'].'.<br>';
  } else {
     // Not logged in
     echo "<h2>Problem:</h2>";
     echo 'You are not logged in.<br><br>';
     echo "<a href=\"index.php\">Login</a><br>";
     echo "</body></html>";
  }
}

function change_password($email, $old_password, $new_password) {
// If old password is correct change password to new_password
// and return true otherwise return false
   if (login($email, $old_password)) {
      $connect=dbconnect();
      $query= "update members set passwd = password('$new_password') ".
              "where email = '$email'";
      $result = mysql_query($query);
      if (!$result) {
        return false;  // not changed
      } else {
        return true;  // changed successfully
      }
   } else {
      return false; // old password was wrong
   }
}

function get_random_word($min_length, $max_length) {
// return a random word of length min to max
   $word = '';
   $dictionary = '/usr/share/dict/words';  // the ispell dictionary
   $fp = fopen($dictionary, 'r');
   if(!$fp) return false;
   $size = filesize($dictionary);

   // go to a random location in dictionary
   srand ((double) microtime() * 1000000);
   $rand_location = rand(0, $size);
   fseek($fp, $rand_location);

   // get the next whole word of the right length in the file
   while (strlen($word)< $min_length||strlen($word)>$max_length||strstr($word,"'")) {
      if (feof($fp))
         fseek($fp, 0);        // if at end, go to start
      $word = fgets($fp, 80);  // skip first word as it could be partial
      $word = fgets($fp, 80);  // the potential password
   };
   $word=trim($word); // trim the trailing \n from fgets
   return $word;
}
```

```
function reset_password($email) {
// set password for user with this email to a random value
// return the new password or false on failure
   $new_password = get_random_word(6, 13);

   if($new_password==false) return false;
   // add a number between 0 and 999 to it
   // to make it a slightly better password
   srand ((double) microtime() * 1000000);
   $rand_number = rand(0, 999);
   $new_password .= $rand_number;

   // encrypt user's password and store it in the database
   $connect = dbconnect();
   $query= "update members set passwd = password('$new_password') ".
           "where email = '$email'";
   $result = mysql_query($query);
   if (!$result)
      return false;  // not changed
   else
      return $new_password;  // successful
}

function notify_password($email, $password) {
// notify the user that their password has been changed
   $from = "From: support@dikaios.org \r\n";
   $mesg = "Your Dikaios ISP password has been changed to $password\n".
           "Please change it the next time you log in.\n";
   if (mail($email, 'Dikaios ISP login information', $mesg, $from))
      return true;
   else
      return false;
}
?>


<!-----------------------------------------------------------
- Filename:  bottom.php
- Written by: Darryl Scroggins
- Last Modified : 2005-04-27 01:28
------------------------------------------------------------>
<html>
<head>
    <? include("header.inc"); ?>
</head>

<body>

<center>
<table>
<tr>
<td>
<h1>
Welcome to Dikaios ISP.<br>
</h1>

To get started select a task from the dropdown above.<br>
Additional dropdowns will be added depending on what you change it to.<br>
<br>

Once you see them set their values and click GO.<br>

</body>
</html>


<!-----------------------------------------------------------
- Filename:  change_passwd2.php
- Written by: Darryl Scroggins
- Last Modified : 2005-04-29 00:59
------------------------------------------------------------>
<?
require_once("authenticate.php");
session_start();
?>
```

```
<html>
<head>
    <? include("header.inc"); ?>
</head>

<body>
<center>
<h2>Changing password</h2>

<?
$email = $HTTP_POST_VARS['email'];
$old_passwd = $HTTP_POST_VARS['old_passwd'];
$new_passwd = $HTTP_POST_VARS['new_passwd'];
$new_passwd2 = $HTTP_POST_VARS['new_passwd2'];

if ($new_passwd!=$new_passwd2)
    die ("Passwords entered were not the same.  Not changed.");

if (strlen($new_passwd)>16 || strlen($new_passwd)<6)
    die ("New password must be between 6 and 16 characters.  Try again.");

// attempt update
if (change_password($email, $old_passwd, $new_passwd)) {
   echo 'Password changed.<br>';
   echo "<a href='isp.php'>Go to ISP Page</a>";
} else {
  echo 'Password could not be changed.';
}
?>
</body>
</html>


<!-----------------------------------------------------------
- Filename:  change_passwd.php
- Written by: Darryl Scroggins
- Last Modified : 2005-04-29 00:56
------------------------------------------------------------>
<?
require_once("authenticate.php");
session_start();
?>

<html>
<head>
   <? require_once("header.inc"); ?>
</head>
<body>

<center>
<h2>Change Password</h2>
<br>

<form action="change_passwd2.php" method=post>
<table>
<tr><td align="right">Email Address: </td>
   <td><input type=text name=email size="30" value="<? echo $email; ?>"></td>
</tr>
<tr><td align="right">Current Password: </td>
    <td><input type=password name=old_passwd size=16 maxlength=16></td>
</tr>
<tr><td align="right">New Password: </td>
    <td><input type=password name=new_passwd size=16 maxlength=16></td>
</tr>
<tr><td align="right">Repeat New Password: </td>
    <td><input type=password name=new_passwd2 size=16 maxlength=16></td>
</tr>
<tr><td colspan="2" align="center">
   <br><input type=submit value="Change Password"></td>
</tr>
</table>

</center>
</body>
</head>
```

```
<!-------------------------------------------------------
- Filename:  cos2.php
- Written by: Darryl Scroggins
- Last Modified : 2005-05-31 00:17
-------------------------------------------------------->
<html>
<head>
    <? include("header.inc"); ?>
</head>

<body>
<?
if (!$studentid || !$gradelevel OR !$semester) {
   die("You must select a student, gradelevel, and semester<br>");
}

require_once('dbconnect.php');
$connect = dbconnect();

function courses($subjectid,$idselected) {
   $s="";
   if ($idselected=="") { $s=" SELECTED"; }
   if ($subjectid) {
      $lable0="---- Select a Course ----";
   } else {
      $lable0="------- Not taking -------";
   }
   $c="<select name=\"classid\" size=\"1\" $s>".
      "<option value=\"\">$lable0</option>\n";

   $q="SELECT c.classid, classname FROM subject_classes s, classes c ".
      "WHERE s.classid=c.classid AND subjectid=\"$subjectid\" ".
      "ORDER BY classname";
   $result = mysql_query($q) or die("Bad query: $q");

   while ($r = mysql_fetch_array($result)){
      $s="";
      $classid = $r["classid"];
      $classname = $r["classname"];
      if ($classid==$idselected) { $s="SELECTED"; }
      $c .= "<option value=\"$classid\"$s>$classname</option>\n";
   }
   $c .= "</select>\n";
   return $c;
}

$query = "SELECT firstname, lastname FROM students WHERE ".
      "studentid=\"$studentid\"";
$result = mysql_query($query) or die("Bad query: $query");
$r = mysql_fetch_array($result);
$studentname=$r["firstname"]." ".$r["lastname"];

$query = "SELECT subject FROM subjects WHERE subjectid=\"$subjectid\"";
$result = mysql_query($query) or die("Bad query: $query");
$r = mysql_fetch_array($result);
$subject=$r["subject"];

if ($cosid) {
   $q="SELECT * FROM courseofstudy WHERE cosid=\"$cosid\"";
   $result = mysql_query($q) or die("Bad query: $q");
   $r = mysql_fetch_array($result);
   $units=$r["units"];
   $hours=$r["hours"];
   $textbooks=$r["textbooks"];
   $tests=$r["tests"];
   $projects=$r["projects"];
   $subjectid=$r["subjectid"];
   $courseid=$r["classid"];
} else { // Set some default values
   $units="5";
   $hours="5";
   $tests="T=\nQ=";
}
?>
<center>
<h2><? echo "Details for the $subject Class to Add<br>"; ?></h2>
<br>
```

```
<form method="POST" action="cos3.php">
<input type="hidden" name="studentid" value="<? echo $studentid; ?>">
<input type="hidden" name="gradelevel" value="<? echo $gradelevel; ?>">
<input type="hidden" name="semester" value="<? echo $semester; ?>">
<input type="hidden" name="subjectid" value="<? echo $subjectid; ?>">

<table align="center">
   <tr><td>Class</td>
       <td>Textbooks</td>
       <td>Tests &amp;amp; Quizzes</td>
       <td>Papers / Projects</td>
   </tr><tr>
      <td valign="top">
      <? echo courses($subjectid,$courseid); ?>
      <br><br>Course Units
      <input type="text" name="units" size="3" value="<?echo $units; ?>">
      <br>Hours per week
      <input type="text" name="hours" size="3" value="<?echo $hours; ?>"></td>
      <td valign="top">
      <textarea name="textbooks" rows="4" cols="30"><?echo $textbooks; ?></textarea>
      </td>
      <td valign="top">
      <textarea name="tests" rows="4" cols="15"><?echo $tests; ?></textarea>
      </td>
      <td valign="top">
      <textarea name="projects" rows="4" cols="30"><?echo $projects; ?></textarea>
      </td>
   </tr><tr><td colspan="2" align="left">
      <br><input type="submit" value="Cancel">
      </td><td colspan="2" align="right">
      <br><input type="submit" value="Update"></td>
   </tr>
</table>
</center>
</form>
</body>
</html>


<!-------------------------------------------------------
- Filename:  cos3.php
- Written by: Darryl Scroggins
- Last Modified : 2005-04-28 20:53
-------------------------------------------------------->
<?
if (!$studentid OR !$gradelevel OR !$semester) {
   die("You must select a student, gradelevel, and semester<br>");
}

if ($submit!="Cancel") {
   require_once('dbconnect.php');
   $connect = dbconnect();
   $query = "SELECT firstname, lastname FROM students WHERE ".
            "studentid=\"$studentid\"";
   $result = mysql_query($query) or die("Bad query: $query");
   $r = mysql_fetch_array($result);
   $studentname=$r["firstname"]." ".$r["lastname"];

   $query="REPLACE courseofstudy (studentid, classid, gradelevel, semester, units, ".
       "hours, textbooks, tests, projects, subjectid) ".
       "VALUES(\"$studentid\",\"$classid\",\"$gradelevel\",\"$semester\",\"$units\",".
       "\"$hours\",\"$textbooks\",\"$tests\",\"$projects\",\"$subjectid\")";
   $result = mysql_query($query) or die("Bad query: $query");
}

header("Location: cos.php?studentid=$studentid&gradelevel=$gradelevel
       &semester=$semester");
?>


<!-------------------------------------------------------
- Filename:  cos.php
- Written by: Darryl Scroggins
- Last Modified : 2005-05-31 00:21
-------------------------------------------------------->
<html>
<head>
```

```
   <? include("header.inc"); ?>
</head>

<body>
<?
   if (!$studentid OR !$gradelevel OR !$semester) {
       echo "You must select a student, gradelevel, and semester<br>";
       die();
   }


require_once('dbconnect.php');
$connect = dbconnect();

function subject_select() {
   $query = "SELECT * FROM subjects ORDER BY subject";
   $result = mysql_query($query) or die("Bad query: $query");
   $subjectselect="<select name=\"subjectid\" size=\"1\">".
       "<option value=\"\" selected>-- Select a subject --</option>\n";

   while ($r = mysql_fetch_array($result)){
      $s="";
      $subject=$r["subject"];
      $subjectid=$r["subjectid"];
      if ($subjectid==$idselected) { $s="SELECTED"; }
      $subjectselect .= "<option value=\"$subjectid\"$s>$subject</option>\n";
    }
    $subjectselect .= "</select>\n";
    return $subjectselect;
}

$query = "SELECT firstname, lastname FROM students WHERE ".
       "studentid=\"$studentid\"";
$result = mysql_query($query) or die("Bad query: $query");
$r = mysql_fetch_array($result);
$studentname=$r["firstname"]." ".$r["lastname"];

?>

<center>
<h2>
Course of Study for
<? echo "$studentname<br>Grade Level $gradelevel - $semester Semester<br>\n"; ?>
</h2>

<?
if (!$printfmt) {
echo "<form method=\"POST\" action=\"cos2.php\">\n".
   "<input type=\"hidden\" name=\"studentid\" value=\"$studentid\">\n".
   "<input type=\"hidden\" name=\"gradelevel\" value=\"$gradelevel\">\n".
   "<input type=\"hidden\" name=\"semester\" value=\"$semester\">\n".
   "<br>To add a class to this course of study select a subject\n".
   "<br>area from the dropdown list \n".
   "and click \"Add Class Details\".<br>\n".
   subject_select().
   " <input type=\"submit\" value=\"Add Class Details\">\n".
   "</form>\n".
   "To display a printable version click <u>\n".
   "<a href=\"cos.php?studentid=$studentid&gradelevel=$gradelevel&semester=$semester&".
   "printfmt=1\">here</a></u> or select the Display Course of Study task.\n";
} else {
   echo "<br>";
}

$query = "SELECT * FROM courseofstudy cos, classes c, subjects s ".
   "WHERE studentid=\"$studentid\" ".
   "AND gradelevel=\"$gradelevel\" AND semester=\"$semester\" ".
   "AND cos.classid=c.classid AND cos.subjectid=s.subjectid ".
   "ORDER BY classname";

$result = mysql_query($query) or die("Bad query: $query");
$classes=mysql_num_rows($result);

if ($classes>0) {
   if ($printfmt) {
       $deletehdr="";
       $edithdr="";
```

```
    } else {
        $deletehdr="<th> \n";
        $edithdr="<th> \n";
    }
    echo "<table align=\"center\" border=\"1\">\n".
        $deletehdr.
        "<th>Subject Area</td>\n".
        "<th>Class\n".
        "<th>Textbooks\n".
        "<th>Tests\n".
        "<th>Papers/Projects\n".
        $edithdr.
        "</tr>\n";
    while ($r = mysql_fetch_array($result)) {
       $cosid=$r['cosid'];
       $subject=$r['subject'];
       $classname=$r['classname'];
       $units=$r['units'];
       $hours=$r['hours'];
       $textbooks=$r['textbooks'];
       $tests=$r['tests'];
       $projects=$r['projects'];
       if ($printfmt) {
          $deleteform="";
          $editform="";
       } else {
          $deleteform="<td valign=\"top\">".
          "<form method=\"POST\" action=\"deleteclasscheck.php\">\n".
          "<input type=\"hidden\" name=\"cosid\" value=\"$cosid\">\n".
          "<input type=\"hidden\" name=\"studentid\" value=\"$studentid\">\n".
          "<input type=\"hidden\" name=\"gradelevel\" value=\"$gradelevel\">\n".
          "<input type=\"hidden\" name=\"semester\" value=\"$semester\">\n".
          "<input type=\"submit\" name=\"delete\"\n".
          "value=\"del\" style=\"font-size:10px\">\n".
          "</form>\n".
          "</td>";
          $editform="<td valign=\"top\">".
          "<form method=\"POST\" action=\"cos2.php\">\n".
          "<input type=\"hidden\" name=\"cosid\" value=\"$cosid\">\n".
          "<input type=\"hidden\" name=\"studentid\" value=\"$studentid\">\n".
          "<input type=\"hidden\" name=\"gradelevel\" value=\"$gradelevel\">\n".
          "<input type=\"hidden\" name=\"semester\" value=\"$semester\">\n".
          "<input type=\"submit\" name=\"edit\"\n".
          "value=\"edit\" style=\"font-size:10px\">\n".
          "</form>\n".
          "</td>";
       }
       echo "<tr>".
       $deleteform .
       "<td align=\"right\" valign=\"top\">".
       $subject.
       "</td><td valign=\"top\">".
       $classname.
       "<br>Units: ".$units." Hours: ".$hours.
       "</td><td valign=\"top\">".
       $textbooks.
       "</td><td valign=\"top\">".
       $tests.
       "</td><td valign=\"top\">".
       $projects.
       "</td>".
       $editform .
       "</tr>\n";
    }
    echo "</table></center>\n";
}

?>
</body>
</html>


<!-------------------------------------------------------
- Filename:  dbconnect.php
- Written by: Darryl Scroggins
- Last Modified : 2005-04-28 20:38
-------------------------------------------------------->
```

```
<?
function dbconnect() {
   // Attempt to make a persistent connection
   $result = mysql_pconnect('localhost', 'defenders', 'secretpassword');
   if (!$result) {
      die("Could not connect to the database server.<br>");
   }
   // Use the dikaios database
   if (!mysql_select_db('dikaios')) {
      die("Could not connect to the Dikaios database.<br>");
   }
   return $result;
}
?>


<!-------------------------------------------------------
- Filename:  deleteclasscheck.php
- Written by: Darryl Scroggins
- Last Modified : 2005-04-16 21:04
-------------------------------------------------------->
<html>
<head>
    <? require_once("header.inc"); ?>
</head>

<body>
<br>
<center>
<table border="0">
<tr><td colspan="2" align="center">
You are about to permanently delete this course and all data associated with it.<br>
Are you sure you want to do this?
</td></tr>
<tr>
<td align="left">
<form method="POST" action="cos.php">
<input type="hidden" name="cosid" value="<? echo $cosid; ?>">
<input type="hidden" name="studentid" value="<? echo $studentid; ?>">
<input type="hidden" name="gradelevel" value="<? echo $gradelevel; ?>">
<input type="hidden" name="semester" value="<? echo $semester; ?>">
<input type="submit" name="goback" value="No Go Back" style="font-size:10px">
</form>
</td>
<td align="right">
<form method="POST" action="deleteclass.php">
<input type="hidden" name="cosid" value="<? echo $cosid; ?>">
<input type="hidden" name="studentid" value="<? echo $studentid; ?>">
<input type="hidden" name="gradelevel" value="<? echo $gradelevel; ?>">
<input type="hidden" name="semester" value="<? echo $semester; ?>">
<input type="submit" name="delete" value="Yes Delete It" style="font-size:10px">
</form>
</td></tr>
</table>
</center>
</body>
</html>


<!-------------------------------------------------------
- Filename:  deleteclass.php
- Written by: Darryl Scroggins
- Last Modified : 2005-04-28 20:56
-------------------------------------------------------->
<?
require_once('dbconnect.php');
$connect = dbconnect();
if ($cosid) {
   $q="DELETE FROM courseofstudy WHERE cosid=\"$cosid\"";
   $result = mysql_query($q);
}
header("Location:
cos.php?studentid=$studentid&gradelevel=$gradelevel&semester=$semester");
?>
```

```
<!-------------------------------------------------------
- Filename:  eval2.php
- Written by: Darryl Scroggins
- Last Modified : 2005-04-28 21:04
-------------------------------------------------------->
<html>
<head>
    <? include("header.inc"); ?>
</head>

<body>
<?
require_once('dbconnect.php');
$connect = dbconnect();

function pack_attendance($a) {
   $n=0;
   for ($i=31; $i>=1; $i--) {
      $n<<=1; //shift n left 1 bit
      if ($a[$i]) { $n+=1; }
   }
   return ($n);
}

function makegradeselect(&$g,$currentgrade) {
    $gradeselect = "<option value=\"\" selected>Grade?</option>";
    $gnum = sizeof($g);
    for ($i=0; $i<$gnum; $i++) {
        $grade=$g[$i];
        $selected="";
        if ($grade==$currentgrade) { $selected="selected"; }
        $gradeselect .= "<option value=\"$grade\" $selected>$grade</option>";
    }
    return $gradeselect;
}


$n=pack_attendance($a);

// Count schooldays
$schooldays=0;
for ($i=1; $i<=31; $i++) {
    if ($a[$i]) { $schooldays++; }
}

$q = "REPLACE monthlyeval SET ".
    "studentid=\"$studentid\", ".
    "gradelevel=\"$gradelevel\",".
    "semester=\"$semester\",".
    "year=\"$year\",".
    "month=\"$month\",".
    "schooldays=\"$schooldays\",".
    "sickdays=\"$sickdays\",".
    "vacationdays=\"$vacationdays\",".
    "attendance=\"$n\",".
    "endofperiod=\"$endofperiod\",".
    "activities=\"$activities\",".
    "fieldtrips=\"$fieldtrips\",".
    "note1=\"$note1\",".
    "note2=\"$note2\",".
    "note3=\"$note3\",".
    "note4=\"$note4\"";

$result = mysql_query($q) or die(mysql_error());
$query = "SELECT grade, points FROM gradepoints ORDER BY points DESC";
$result = mysql_query($query) or die("Bad query: $query");
while ($r = mysql_fetch_array($result)){
    $grades[] = $r["grade"];
    $points[$r["grade"]] = $r["points"];
}

$q = "SELECT c.classid, classname FROM courseofstudy cos, classes c ".
     "WHERE studentid=\"$studentid\" ".
     "AND gradelevel=\"$gradelevel\" AND semester=\"$semester\" ".
     "AND cos.classid=c.classid ".
     "ORDER BY classname";
```

```
$result = mysql_query($q) or die(mysql_error());
?>

<center>
<form method="POST" action="eval3.php">
<input type="hidden" name="studentid" value="<? echo $studentid; ?>">
<input type="hidden" name="gradelevel" value="<? echo $gradelevel; ?>">
<input type="hidden" name="semester" value="<? echo $semester; ?>">
<input type="hidden" name="month" value="<? echo $month; ?>">
<input type="hidden" name="year" value="<? echo $year; ?>">

<br><br><table border="1">
<th>Subject<th>Grade<th>Pages or<br>Units<th>Projects & Comments

<?
$i=0;
while ($r = mysql_fetch_assoc($result)) {
    $classid= $r["classid"];
    $classname= $r["classname"];
    $q="SELECT * FROM coursework WHERE studentid=$studentid AND ".
       "year=$year AND month=$month AND classid=$classid";
    $result2 = mysql_query($q) or die(mysql_error());
    $nrows=mysql_num_rows($result2);
    if ($nrows) {
        $r2=mysql_fetch_assoc($result2);
        $grade=$r2["grade"];
        $pages=$r2["pages"];
        $projects=$r2["projects"];
    } else {
        $grade="";$pages="";$projects="";
    }

    //Set the grade to whatever was in the database
    // $gradeselect=makegradeselect(&$grades,$grade);
    $gradeselect="<select name=\"grade[]\" size=\"1\">".
       makegradeselect(&$grades,$grade). "</select>";


    echo "<tr><td>".$classname.
    "<input type=\"hidden\" name=\"classid[$i]\" value=\"$classid\">".
    "</td><td>".
    $gradeselect.
    "</td><td>".
    "<textarea rows=\"2\" cols=\"20\" name=\"pages[$i]\">$pages</textarea>".
    "</td><td>".
    "<textarea rows=\"2\" cols=\"30\" name=\"projects[$i]\">$projects</textarea>".
    "</td></tr>";
    $i++;
}
?>

</table>
<input type="hidden" name="classes" value="<? echo $i; ?>">
<br><input type="submit" value="Submit">
</form>
</center>
</body>
</html>


<!-----------------------------------------------------------
- Filename:  eval3.php
- Written by: Darryl Scroggins
- Last Modified : 2005-05-30 22:20
------------------------------------------------------------>
<?
require_once('dbconnect.php');
$connect = dbconnect();

for ($i=0; $i<$classes; $i++) {
$q = "REPLACE coursework SET ".
    "studentid=\"$studentid\", ".
    "year=\"$year\",".
    "month=\"$month\",".
    "classid=\"$classid[$i]\",".
    "grade=\"$grade[$i]\",".
    "pages=\"$pages[$i]\",".
```

```
    "projects=\"$projects[$i]\"";

    $result = mysql_query($q) or die(mysql_error());
}
header("Location:
showeval.php?studentid=$studentid&gradelevel=$gradelevel&year=$year&month=$month");
?>


<!-------------------------------------------------------
- Filename:  eval.php
- Written by: Darryl Scroggins
- Last Modified : 2005-04-28 21:03
-------------------------------------------------------->
<html>
<head>
    <? include("header.inc"); ?>
</head>

<body>
<?
require_once('dbconnect.php');
$connect = dbconnect();

function unpack_attendance($n,&$a) {
   $mask=1;      //right most bit set
   for ($i=1; $i<=31; $i++) {
      $a[$i]=($n & $mask) ? 1:0;
      $mask<<=1; //shift the mask left 1 bit
   }
}

$q="SELECT * FROM monthlyeval WHERE studentid=$studentid AND year=$year".
   " AND month=$month";

$result = mysql_query($q) or die(mysql_error());
$ncols=mysql_num_fields($result);
$nrows=mysql_num_rows($result);
$newrec=1;
if ($nrows) {
    //editing data
    $newrec=0;
    $r = mysql_fetch_assoc($result);
    $schooldays= $r["schooldays"];
    $sickdays= $r["sickdays"];
    $vacationdays= $r["vacationdays"];
    $attendance= $r["attendance"];
    unpack_attendance($attendance,&$a);
    $activities= $r["activities"];
    $fieldtrips= $r["fieldtrips"];
    $note1= $r["note1"];
    $note2= $r["note2"];
    $note3= $r["note3"];
    $note4= $r["note4"];
}
?>

<center><h1>Monthly Evaluation for <?echo $month."/".$year; ?></h1><br>
<form method="POST" action="eval2.php">

<input type="hidden" name="studentid" value="<? echo $studentid; ?>">
<input type="hidden" name="gradelevel" value="<? echo $gradelevel; ?>">
<input type="hidden" name="semester" value="<? echo $semester; ?>">
<input type="hidden" name="month" value="<? echo $month; ?>">
<input type="hidden" name="year" value="<? echo $year; ?>">

<?

$daysinmonth=date("t",mktime(0,0,0,$month));

$cal="<table border=1>\n";
$cal.="<tr><th>S<th>M<th>T<th>W<th>T<th>F<th>S<tr>";

for ($day=1;$day<=$daysinmonth;$day++) {
        $dow=date("w",mktime(0,0,0,$month,$day,$year));
        if ($day==1) {
                for ($i=0;$i<$dow;$i++) {
```

```
                        $cal.="<td> </td>";
                }
        }
        if ($dow==0) { $cal.="</tr>\n<tr>"; }
        if ($newrec) {
                if ($dow>0 AND $dow <6) { $c="checked"; } else { $c=""; }
        } else {
                if ($a[$day]) { $c="checked"; } else { $c=""; }
        }
        $cal.="<td align=right>$day ".
           "<input type=\"checkbox\" name=\"a[$day]\" value=\"1\" $c ></td>";
        if($day==$daysinmonth) {
                $leftover=6-$dow;
                for ($i=0;$i<$leftover;$i++) {
                        $cal.="<td> </td>";
                }
                $cal.="</tr>\n";
        }
}
$cal.="</table>\n";
?>

<table border="0">
<tr><td align="center" colspan="2">Please enter the number of days missed
    and check each day in attendance.</td></tr>
<tr><td align="center" valign="top">
<br>
Sick Days: 
<input type="text" name="sickdays" size="4" value="<? echo $sickdays; ?>">
<br>
<br>
Vacation Days: 
<input type="text" name="vacationdays" size="4" value="<? echo $vacationdays; ?>">
</td><td><? echo $cal; ?></td></tr>
</table>

<br>
<table>
<tr><td colspan="5" align="center">Please list activities, fieldtrips, and any other
educational experiences that occurred this month.</td></tr>
<tr>
        <td align="right" valign="top">Activities: </td>
        <td><textarea rows="2" cols="50" name="activities"><? echo $activities; ?
></textarea></td>
        <td width="5%"> </td>
        <td align="right" valign="top">Note2: </td>
        <td><textarea rows="2" cols="50" name="note2"><? echo $note2; ?></textarea></td>
</tr><tr>
        <td align="right" valign="top">Fieldtrips:</td>
        <td><textarea rows="2" cols="50" name="fieldtrips"><? echo $fieldtrips; ?
></textarea></td>
        <td width="5%"> </td>
        <td align="right" valign="top">Note3:</td>
        <td><textarea rows="2" cols="50" name="note3"><? echo $note3; ?></textarea></td>
</tr><tr>
        <td align="right" valign="top">Note1:</td>
        <td><textarea rows="2" cols="50" name="note1"><? echo $note1; ?></textarea></td>
        <td width="5%"> </td>
        <td align="right" valign="top">Note4:</td>
        <td><textarea rows="2" cols="50" name="note4"><? echo $note4; ?></textarea></td>
</tr><tr>
        <td colspan="5" align="center"><input type="submit" value="Submit"></td>
</tr>
</table>
</form>

</body>
</html>


<!-------------------------------------------------------
- Filename:  get_passwd2.php
- Written by: Darryl Scroggins
- Last Modified : 2005-04-29 00:51
-------------------------------------------------------->
<html>
<head>
```

```
   <? include("header.inc"); ?>
</head>

<body>
<center>
<h2>Resetting password</h2>
<?
require_once("authenticate.php");
$email = $HTTP_POST_VARS['email'];

$msg=register($email);
if ($msg=="success") {
   echo "Your new password has been sent to $email.";
} else {
   echo $msg;
}
echo "<br><br><a href='index.php'>Go to Login Page</a>";
?>
</body>
</html>


<!-------------------------------------------------------
- Filename:  get_passwd.php
- Written by: Darryl Scroggins
- Last Modified : 2005-04-29 00:52
-------------------------------------------------------->
<html>
<head>
    <? require_once("header.inc"); ?>
</head>

<body>
<center>
<h2>Getting a New Password</h2>
<br>
<form action="get_passwd2.php" method=post>
<table>
   <tr><td>
       Enter your email and click "Send Password"
   </td></tr>
   <tr><td>
      <input type=text name=email size=35 maxlength=35 value="<? echo $email; ?>">
   </td></tr>
   <tr><td align="center">
      <input type=submit value="Send Password">
   </td></tr>
</table>
</center>
</body>
</html>


<!-------------------------------------------------------
- Filename:  header.inc
- Written by: Darryl Scroggins
- Last Modified : 2004-05-14 00:45
-------------------------------------------------------->
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<meta name="Keywords" content="telecom, datacom, equipment, scrap, reuse">
<title>Dikaios Christian Home Educators</title>
<STYLE TYPE="text/css">
       a:link { color:#000000; text-decoration:none; }
       a:visited { color:#000000; text-decoration:none; }
       a:hover { color:#B13B30; text-decoration:none; }
       a:active { color:#715454; text-decoration:none; }
       a.body:link { color:#0000CF; }
       a.body:visited { color:#0000CF; }
       a.body:hover { color:#CF0000 }
       body { color:#000000; background:#FFF2E2 }
       table { empty-cells:show }
       table.border { width:70%; border: solid black; border-width:thick;
                      border-style: groove; padding:1em; background:#C9C1B2; }
</STYLE>
```

```
<!-------------------------------------------------------
- Filename:  index2.php
- Written by: Darryl Scroggins
- Last Modified : 2005-05-31 00:25
-------------------------------------------------------->
<? require_once('valid_user.php'); ?>
<html>
<head>
   <? include("header.inc"); ?>
      <script type="text/javascript">
          function resubmit() {
              document.forms[0].action="index2.php";
              document.forms[0].target="frame1";
              document.forms[0].submit();
          }
      </script>
</head>

<body>
<center>

<?
require_once('dbconnect.php');
$connect = dbconnect();

$memberid=$_SESSION['member_id'];

$query = "SELECT studentid, firstname FROM students ".
    "WHERE memberid=\"$memberid\" ORDER BY firstname";
$result = mysql_query($query) or die("Bad query: $query");
$students = "<option value=\"\" selected>- Student? -</option>";
while ($r = mysql_fetch_array($result)){
   $sid = $r["studentid"];
   $student = $r["firstname"];
   $selected="";
   if ($studentid==$sid) { $selected="selected"; }  //studentid is the select name
   $students .= "<option value=\"$sid\" $selected>$student</option>";
}

function make_option($prev_selection, $set_value, $prompt) {
   $selected="";
   if ($prev_selection==$set_value) { $selected="selected"; }
   return "<option value=\"$set_value\" $selected>$prompt</option>";
}

if ($funcname) { $frame="frame2"; } else { $frame="frame1"; }
echo "<form method=\"POST\" action=\"$funcname\" target=\"$frame\">";

//TASKS
echo "<select name=\"funcname\" size=\"1\" onChange=\"resubmit()\">";
   echo make_option($funcname,"","- Select a Task -");
   echo make_option($funcname,"cos.php","Prepare Course of Study");
   echo make_option($funcname,"showcos.php","Display Course of Study");
   echo make_option($funcname,"eval.php","Prepare Monthly Evaluation");
   echo make_option($funcname,"showeval.php","Display Monthly Evaluation");
   echo make_option($funcname,"reportcard.php","Prepare Semester Report Card");
   echo make_option($funcname,"showreportcard.php","Display Semester Report Card");
   echo make_option($funcname,"showgradplan.php","Display High School Graduation Plan");
   echo make_option($funcname,"showtranscript.php","Display High School Transcript");
echo "</select> ";

if ($funcname) {
echo "<select name=\"studentid\" size=\"1\">";
       echo $students;
echo "</select> ";

// All tasks need gradelevel and semester except these
if ($funcname!="showgradplan.php" AND $funcname!="showtranscript.php") {
   echo "<select name=\"gradelevel\" size=\"1\">";
   echo make_option($gradelevel,"","- Grade? -");
   echo make_option($gradelevel,"12","12");
   echo make_option($gradelevel,"11","11");
   echo make_option($gradelevel,"10","10");
   echo make_option($gradelevel,"9","9");
   echo make_option($gradelevel,"8","8");
   echo make_option($gradelevel,"7","7");
   echo make_option($gradelevel,"6","6");
```

```
   echo make_option($gradelevel,"5","5");
   echo make_option($gradelevel,"4","4");
   echo make_option($gradelevel,"3","3");
   echo make_option($gradelevel,"2","2");
   echo make_option($gradelevel,"1","1");
   echo make_option($gradelevel,"K","K");
   echo "</select> ";

   echo "<select name=\"semester\" size=\"1\">";
   echo make_option($semester,"","- Semester? -");
   echo make_option($semester,"First","First");
   echo make_option($semester,"Second","Second");
   echo "</select>";
}

// These functions need to know the month and year
if ($funcname=="eval.php" OR $funcname=="showeval.php") {
   if ($month=="") {
        $lastmonth = mktime (0,0,0,date("m")-1,date("d"),date("Y"));
        $month=date("m",$lastmonth);
        $year=date("Y",$lastmonth);
   }
   echo "<select name=\"month\" size=\"1\">";
   echo make_option($month,"","- Month? -");
   echo make_option($month,"01","January");
   echo make_option($month,"02","February");
   echo make_option($month,"03","March");
   echo make_option($month,"04","April");
   echo make_option($month,"05","May");
   echo make_option($month,"06","June");
   echo make_option($month,"07","July");
   echo make_option($month,"08","August");
   echo make_option($month,"09","September");
   echo make_option($month,"10","October");
   echo make_option($month,"11","November");
   echo make_option($month,"12","December");
   echo "</select> ";

   echo "<select name=\"year\" size=\"1\">";
   if ($year=="") {
        $year=date("Y");
   }
   echo make_option($year,"","- Year? -");
   echo make_option($year,"2003","2003");
   echo make_option($year,"2004","2004");
   echo make_option($year,"2005","2005");
   echo make_option($year,"2006","2006");
   echo make_option($year,"2007","2007");
   echo make_option($year,"2008","2008");
   echo "</select>";
}

echo " <input type=\"submit\" value=\" GO \">";
}
?>
</form>
</center>
</body>
</html>


<!-------------------------------------------------------
- Filename:  index.php
- Written by: Darryl Scroggins
- Last Modified : 2005-05-31 00:29
-------------------------------------------------------->
<html>
<head>
    <? require_once("header.inc"); ?>
</head>

<body>

<center>
<br>
<h1>Welcome to Dikaios ISP</h1>
<h2>Please Login</h2>
```

```
<form method=post action="member.php">
<table>
<tr>
<td>Email Address:<br>
<input type=text name=email size="30" value="<? echo $email; ?>"></td></tr>
<tr>
<td>Password:<br>
<input type=password name=passwd size="16"> 
<input type=submit value="Log in">
</td></tr>
<tr>
<td><a href="get_passwd.php?email=<? echo $email; ?>">Get a new password.</a>
</td>
</tr>
<tr>
<td>
<a href="change_passwd.php?email=<? echo $email; ?>">Change my password.</a>
</td>
</tr>
<tr>
<td>
<?
if ($badlogin) {
   echo "<font color='RED'>";
   echo "Your email address and/or password are incorrect!<br>";
   echo "If you think the email address is correct try to get ";
   echo "a new password!</font><br>";
}
?>
</td>
</tr>
</table>
</form>

</center>
</body>
</html>


<!-------------------------------------------------------
- Filename:  isp.php
- Written by: Darryl Scroggins
- Last Modified : 2005-05-31 00:29
-------------------------------------------------------->
<? require_once('valid_user.php'); ?>
<html>
<head>
   <? include("header.inc"); ?>
</head>

<frameset rows="45,*" frameborder=1 bordercolor="#FFF2E2">
   <frame src="index2.php" name="frame1">
   <frame src="bottom.php" name="frame2">
   <noframes><a href="index2.html">Main Menu</a></noframes>
</frameset>
</html>


<!-------------------------------------------------------
- Filename:  member.php
- Written by: Darryl Scroggins
- Last Modified : 2005-04-29 00:55
-------------------------------------------------------->
<?
require_once('authenticate.php');
session_start();

$email = $HTTP_POST_VARS['email'];
$passwd = $HTTP_POST_VARS['passwd'];

$valid_login=FALSE;
if ($email AND $passwd) {
   if ($memberid=login($email, $passwd)) {
      $_SESSION['valid_user'] = $email;
      $_SESSION['member_id'] = $memberid;
      $valid_login=TRUE;
   }
```

```
}

if ($valid_login) {
   header("Location: isp.php");
} else {
   header("Location: index.php?email=$email&badlogin=1");
}
?>


<!-------------------------------------------------------
- Filename:  admin/membershiplabels.php
- Written by: Darryl Scroggins
- Last Modified : 2005-05-31 00:05
-------------------------------------------------------->
<?
include ("class.label.php");
require_once('../dbconnect.php');
$connect = dbconnect();

$q="SELECT title,firstname,lastname,address,city,state,zipcode ".
   "FROM members ORDER BY lastname";

$result = mysql_query($q);
$nrows=mysql_num_rows($result);
if ($nrows) {
   $labeltype="Av5160"; //this is an Avery label used for mailing labels
   $label= new Clabel($labeltype);
   /*
   The Av5160 is expecting an array of four values
   array ('line1'=> first line of address label (typically first and last name)
          'line2'=> second line of address label (typically streeet address)
          'line3'=> third line of address label (typically PO BOX)
          'line4'=> fourth line of address label (typically city,state zip)
         )
   the third line is optional and if it is not present, the class will omit it and
   move the fourth line up on the label
   */
   while ($r = mysql_fetch_assoc($result)) {
       $line1= trim($r["title"] . " " . $r["firstname"]) . " " . $r["lastname"];
       $line2= $r["address"];
       $line3= $r["city"] . ", " . $r["state"] . " " . $r["zipcode"];
       $info[]=array('line1'=>$line1,'line2'=>$line2,'line3'=>$line3,'line4'=>$line4);
   }
   reset($info);
   $label->makeLabel($info);
}
?>


<!-------------------------------------------------------
- Filename:  reportcard2.php
- Written by: Darryl Scroggins
- Last Modified : 2005-04-28 21:15
-------------------------------------------------------->
<?
if (!$studentid || !$gradelevel OR !$semester) {
   echo "You must select a student, gradelevel, and semester<br>";
   die ("Click the Back Button and try again.");
}

require_once('dbconnect.php');
$connect = dbconnect();

$gnum=sizeof($grade);
for ($i=0; $i<$gnum; $i++) {
    $query="UPDATE courseofstudy SET classgrade=\"$grade[$i]\", ".
       "unitscompleted=\"$uc[$i]\" WHERE cosid=\"$cosid[$i]\" LIMIT 1";
    $result = mysql_query($query) or die("Bad query: $query");
}
header("Location: reportcard.php?studentid=$studentid&gradelevel=$gradelevel
       &semester=$semester&printfmt=1");
?>


<!-------------------------------------------------------
- Filename:  reportcard.php
```

```
- Written by: Darryl Scroggins
- Last Modified : 2005-04-28 21:14
------------------------------------------------------------>
<html>
<head>
    <? include("header.inc"); ?>
</head>

<body>
<?
if (!$studentid OR !$gradelevel OR !$semester) {
    die("You must select a student, gradelevel, and semester<br>");
}

require_once('dbconnect.php');
$connect = dbconnect();

$query= "SELECT firstname, lastname FROM students WHERE ".
        "studentid=\"$studentid\"";
$result=mysql_query($query) or die("Bad query: $query");
$r=mysql_fetch_array($result);
$studentname=$r["firstname"]." ".$r["lastname"];

$query = "SELECT grade, points FROM gradepoints ORDER BY points DESC";
$result = mysql_query($query) or die("Bad query: $query");
while ($r = mysql_fetch_array($result)){
    $grades[] = $r["grade"];
    $points[$r["grade"]] = $r["points"];
}

function makegradeselect(&$g,$currentgrade) {
    $gradeselect = "<option value=\"\" selected>Grade?</option>";
    $gnum = sizeof($g);
    for ($i=0; $i<$gnum; $i++) {
        $grade=$g[$i];
        $selected="";
        if ($grade==$currentgrade) { $selected="selected"; }
        $gradeselect .= "<option value=\"$grade\" $selected>$grade</option>";
    }
    return $gradeselect;
}

// Tab is used for spacing and to make underlines
$tab="        ";
?>

<center>
<table><tr><td>
<?
echo "<center><h2><i>Dikaios Christian Academy</i></h2></center>\n".
     "<h3>Report Card for: <u>  $studentname  </u>   \n".
     "Grade Level: <u>  $gradelevel - $semester semester </u></h3>\n";

if (!$printfmt) {
echo "<form method=\"POST\" action=\"reportcard2.php\">\n".
   "<input type=\"hidden\" name=\"studentid\" value=\"$studentid\">\n".
   "<input type=\"hidden\" name=\"gradelevel\" value=\"$gradelevel\">\n".
   "<input type=\"hidden\" name=\"semester\" value=\"$semester\">\n";
} else {
    $query2 = "SELECT activities,fieldtrips,note1,note2,note3,note4 ".
    " FROM monthlyeval WHERE studentid=\"$studentid\" ".
    "AND gradelevel=\"$gradelevel\" AND semester=\"$semester\"";
    $result2 = mysql_query($query2) or die("Bad query: $query");
}

$query = "SELECT * FROM courseofstudy cos, classes c, subjects s ".
    "WHERE studentid=\"$studentid\" ".
    "AND gradelevel=\"$gradelevel\" AND semester=\"$semester\" ".
    "AND cos.classid=c.classid AND cos.subjectid=s.subjectid ".
    "ORDER BY classname";

$result = mysql_query($query) or die("Bad query: $query");
$classcount=mysql_num_rows($result);

if ($classcount>0) {
echo "<table width=\"100%\" align=\"center\" border=\"1\">\n".
    "<th>Subject Area</th>\n".
```

```
"<th>Class</th>\n".
"<th>Hours</th>\n".
"<th>Units<br>Attempted</th>\n".
"<th>Units<br>Completed</th>\n".
"<th>Grade</th>\n".
"</tr>\n";
$totalunits=0; $gpa=0;
$totalhours=0; $totalunitsattempted=0;
while ($r = mysql_fetch_array($result)) {
    $cosid=$r['cosid'];
    $subject=$r['subject'];
    $classname=$r['classname'];
    $units=$r['units'];
    $hours=$r['hours'];
    $classgrade=$r['classgrade'];
    $unitscompleted=$r['unitscompleted'];
    if (!$printfmt) {
       if ($unitscompleted=="") {
           $unitscompleted=$units;
       };
       $unitsstr="<input type=\"text\" name=\"uc[]\" size=\"4\"".
                 " value=\"$unitscompleted\">";
       $unitscompleted=$unitsstr;

       //Set the grade to whatever was in the database
       $gradeselect=makegradeselect(&$grades,$classgrade);
       $classgrade="<select name=\"grade[]\" size=\"1\">".
       $gradeselect. "</select>";
    } else {
        $totalhours+=$hours;
        $totalunitsattempted+=$units;
       $totalunits+=$unitscompleted;
       $gpa+=$unitscompleted * $points[$classgrade];
    }
echo "<tr>".
    "<td align=\"right\" valign=\"top\">".
           "<input type=\"hidden\" name=\"cosid[]\" value=\"$cosid\">".
            $subject.
        "</td>\n<td valign=\"top\">".
           $classname.
        "</td>\n<td valign=\"top\" align=\"right\">".
           $hours.
        "</td>\n<td valign=\"top\" align=\"right\">".
           $units." ".
        "</td>\n<td valign=\"top\" align=\"right\">".
            $unitscompleted." ".
        "</td>\n<td valign=\"top\" align=\"center\">".
            " ".$classgrade.
    "</td>\n".
    "</tr>\n";
}

if ($printfmt) {
   if ($totalunits>0) { $gpa /= $totalunits; } else { $gpa = 0; }
   echo "<tr><td colspan=\"2\" align=\"right\" >Totals: </td>".
     "<td align=\"right\">".number_format($totalhours,2)."</td>".
     "<td align=\"right\">".number_format($totalunitsattempted,2)."</td>".
     "<td align=\"right\">".number_format($totalunits,2)."</td>".
    "<td align=\"center\">".number_format($gpa,2)."</td></tr>";
   echo "<tr><td colspan=\"6\">".
     "<center><b>EXTRA-CURRICULAR ACTIVITIES</b></center></td></tr>".
    "<tr><td colspan=\"6\">";
   $extra="";
   while ($r2 = mysql_fetch_array($result2)) {
       $act=$r2['activity'];
       $trips=$r2['fieldtrips'];
       $n1=$r2['note1'];
       $n2=$r2['note2'];
       $n3=$r2['note3'];
       $n4=$r2['note4'];
       if ($act) { $extra.= $act."<br>"; }
       if ($trips) { $extra.= $trips."<br>"; }
       if ($n1) { $extra.= $n1."<br>"; }
       if ($n2) { $extra.= $n2."<br>"; }
       if ($n3) { $extra.= $n3."<br>"; }
       if ($n4) { $extra.= $n4."<br>"; }
   }
```

```
        if ($extra) {
            echo $extra;
        } else {
            echo "None<br>";
        }
        echo "</td></tr>";
        echo "</table>\n";
        echo "<br>Teacher's Signature: <u>".$tab.$tab.$tab.$tab.$tab.$tab.$tab.
          "</u> Date: <u>".$tab.$tab.$tab."</u><br>\n";
        echo "<br>ISP Admin's Signature: <u>".$tab.$tab.$tab.$tab.$tab.$tab.$tab.
          "</u> Date: <u>".$tab.$tab.$tab."</u><br>\n";
    } else {
        echo "</table>\n".
          "<br>Please enter grades and units completed and click \"Save Grades\".".
          " <input type=\"submit\" value=\"Save Grades\"><br><br>\n".
          "To display a printable version click <u>".
          "<a href=\"reportcard.php?studentid=$studentid&gradelevel=$gradelevel
          "&semester=$semester&".
          "printfmt=1\">here</a></u> or select the Display Report Card task.\n";
    }
}

if (!$printfmt) {
    echo "</form>\n";
}

?>
</td></tr></table></center>
</body>
</html>


<!-------------------------------------------------------
- Filename:  showcos.php
- Written by: Darryl Scroggins
- Last Modified : 2004-10-24 23:22
-------------------------------------------------------->
<?
header("Location: cos.php?studentid=$studentid&gradelevel=$gradelevel
       &semester=$semester&printfmt=1");
?>


<!-------------------------------------------------------
- Filename:  showeval.php
- Written by: Darryl Scroggins
- Last Modified : 2005-04-28 21:19
-------------------------------------------------------->
<html>
<head>
    <? include("header.inc"); ?>
</head>

<body>

<?
if (!$studentid OR !$gradelevel OR !$year OR !$month) {
   die("You must select a student, gradelevel, year, and month<br>");
}

require_once('dbconnect.php');
$connect = dbconnect();

function unpack_attendance($n,&$a) {
   $mask=1;      //right most bit set
   for ($i=1; $i<=31; $i++) {
      $a[$i]=($n & $mask) ? 1:0;
      $mask<<=1; //shift the mask left 1 bit
   }
}

$q="SELECT * FROM monthlyeval WHERE studentid=$studentid AND ".
   "$gradelevel=$gradelevel AND year=$year AND month=$month";

$result = mysql_query($q) or die(mysql_error());
$nrows=mysql_num_rows($result);
if ($nrows) {
```

```
    $r = mysql_fetch_assoc($result);
    $gradelevel=$r["gradelevel"];
    $schooldays= $r["schooldays"];
    $sickdays= $r["sickdays"];
    $vacationdays= $r["vacationdays"];
    $attendance= $r["attendance"];
    $activities=$r["activities"];
    $fieldtrips=$r["fieldtrips"];
    $note1=$r["note1"];
    $note2=$r["note2"];
    $note3=$r["note3"];
    $note4=$r["note4"];
    unpack_attendance($attendance,&$a);
}

$q="SELECT sum(schooldays) as totalschooldays FROM monthlyeval ".
   "WHERE studentid=$studentid AND
concat(year,month)<=\"$year$month\"";
$result = mysql_query($q) or die(mysql_error());
$nrows=mysql_num_rows($result);
if ($nrows) {
    $r = mysql_fetch_assoc($result);
    $totalschooldays= $r["totalschooldays"];
}

// Get student name from studentid
$q="SELECT firstname, lastname FROM students WHERE studentid=$studentid";
$result = mysql_query($q) or die(mysql_error());
$r = mysql_fetch_assoc($result);
$firstname=$r["firstname"];
$lastname=$r["lastname"];

?>

<center>
<font size="+2">
<b><i>Dikaios Christian Academy</i></b><br>
<b>Monthly Evaluation Form</b><br>
</font>
</center>
<br>
<br>
<center>
<table>
<tr>
<td>
Name: <u><?echo " ".$firstname." ".$lastname." "; ?></u>
  Grade Level: <u><?echo " ".$gradelevel." "; ?></u>
  Date: <u><?echo " ".$month."/".$year." "; ?></u><br>
<br>

School Days: <u> <? echo $schooldays; ?> </u>  
Sick Days: <u> <? echo $sickdays; ?> </u>  
Vacation Days: <u> <? echo $vacationdays; ?> </u>  
Total School Days: <u> <? echo $totalschooldays; ?> </u>
<br><br>
</td></tr>
<tr><td>

<table border='1' align='center'><tr>

<?
for ($i=1; $i<=31; $i++) {
    $checked= (($a[$i]) ? "checked1.png":"unchecked1.png");
    echo "<td align='center' valign='top'>$i<br><img src=\"$checked\"></td>";
    if ($i==16) { echo "</tr><tr>"; }
}
?>

<td> </td></tr></table>
<br>
</td></tr>
<tr><td>
<table border="1" width="100%">
<th>Subject</th><th>Grade</th><th>Pages or Units</th>
<th>Projects &amp;amp; Comments</th>
<?
```

```
$q="SELECT * FROM coursework cw, classes c WHERE studentid=$studentid AND ".
   "year=$year AND month=$month AND cw.classid=c.classid ORDER BY classname";
$result2 = mysql_query($q) or die(mysql_error());

while ($r2 = mysql_fetch_assoc($result2)) {
   $classname=$r2["classname"];
   $grade=$r2["grade"];
   $pages=$r2["pages"];
   $projects=$r2["projects"];
   echo "<tr><td valign='top'>$classname</td>".
        "<td align='center' valign='top'>$grade</td>".
        "<td valign='top'>$pages</td>".
        "<td valign='top'>$projects</td></tr>";
}
?>

</table>
<br>
</td></tr>
<tr><td align="center">
Extra-curricular Activities
<table border="1" width="100%">
   <tr>
        <td align="right" valign="top">Activities: </td>
        <td width="90%" valign="top"><? echo $activities; ?></td>
   </tr><tr>
        <td align="right" valign="top">Fieldtrips:</td>
        <td width="90%" valign="top"><? echo $fieldtrips; ?></td>
   </tr><tr>
        <td align="right" valign="top">Note1:</td>
        <td width="90%" valign="top"><? echo $note1; ?></td>
   </tr><tr>
        <td align="right" valign="top">Note2: </td>
        <td width="90%" valign="top"><? echo $note2; ?></td>
   </tr><tr>
        <td align="right" valign="top">Note3:</td>
        <td width="90%" valign="top"><? echo $note3; ?></td>
   </tr><tr>
        <td align="right" valign="top">Note4:</td>
        <td width="90%" valign="top"><? echo $note4; ?></td>
   </tr>
</table>
</td>
</tr>
</table>
</body>
</html>


<!-------------------------------------------------------
- Filename:  showgradplan.php
- Written by: Darryl Scroggins
- Last Modified : 2005-04-28 21:21
-------------------------------------------------------->
<html>
<head>
    <? include("header.inc"); ?>
</head>

<body>
<?
if (!$studentid) {
   die("You must select a student<br>");
}

require_once('dbconnect.php');
$connect = dbconnect();

$query = "SELECT firstname, lastname FROM students WHERE ".
       "studentid=\"$studentid\"";
$result = mysql_query($query) or die("Bad query: $query");
$r = mysql_fetch_array($result);
$studentname=$r["firstname"]." ".$r["lastname"];
$grandtotalplanned=0;
$grandtotalrequired=0;
?>
```

```
<center>
<table><tr><td>
<?
echo "<center><h2><i>Dikaios Christian Academy</i></h2>".
     "<h3>High School Graduation Plan for: <u>  $studentname  </u>".
     "</h3></center>\n";

$query = "SELECT subjectid, subject, unitsrequired FROM subjects ".
    "WHERE unitsrequired > 0.0 ORDER BY printorder";
$result = mysql_query($query) or die("Bad query: $query");
echo "<center><font size=\"-2\">";
echo "<table border=\"1\"><tr>\n".
     "<th rowspan=\"2\">Subject Area</th>\n".
     "<th colspan=\"2\">9th Grade</th>\n".
     "<th colspan=\"2\">10th Grade</th>\n".
     "<th colspan=\"2\">11th Grade</th>\n".
     "<th colspan=\"2\">12th Grade</th>\n".
     "<th colspan=\"2\">Total Units</th>\n".
     "</tr>\n".
     "<tr>".
     "<th>1st Sem</th>".
     "<th>2nd Sem</th>".
     "<th>1st Sem</th>".
     "<th>2nd Sem</th>".
     "<th>1st Sem</th>".
     "<th>2nd Sem</th>".
     "<th>1st Sem</th>".
     "<th>2nd Sem</th>".
     "<th>Planned</th>".
     "<th>Required</th>".
     "</tr>";

while ($r = mysql_fetch_array($result)){
    $sid = $r["subjectid"];
    $subject = $r["subject"];
    $unitsrequired = $r["unitsrequired"];
    $totalplanned=0;

    echo "<tr>\n".
       "<td>".$subject."</td>\n";

    $query2 = "SELECT subjectid,cos.classid,units,gradelevel,semester, ".
        "classname FROM courseofstudy cos, classes ".
        "WHERE studentid=\"$studentid\" AND subjectid=\"$sid\" ".
       "AND cos.classid=classes.classid ".
       "ORDER BY gradelevel,semester";

    $result2 = mysql_query($query2) or die("Bad query: $query2");
    for ($sem=1;$sem<=8;$sem++) {
        $classes[$sem]="";
    }

    while ($r2 = mysql_fetch_array($result2)) {
       $units=$r2["units"];
       $classname=$r2["classname"];
        $totalplanned += $units;
       $yr=$r2["gradelevel"]+0;
       $s=($r2["semester"]=="Second")+1; //Should be able to get the enum val
        $sem=($yr-9)*2+$s; //Should be 1 to 8
        $classes[$sem].= $classname."<br>".$units." units<br>";
    }

    for ($sem=1;$sem<=8;$sem++) {
       echo "<td>".$classes[$sem]."</td>\n";
    }
    echo "<td align=\"right\">".number_format($totalplanned,1)."</td>\n".
         "<td align=\"right\">".number_format($unitsrequired,1)."</td>\n".
    "</tr>\n";
    $grandtotalplanned+=$totalplanned;
    $grandtotalrequired+=$unitsrequired;
}

echo "<tr><td colspan=\"7\">Note that any units taken in a specific subject area ".
     "above the requirement may be counted as electives. ".
     "This means that the number of elective units required do not have to be met ".
     "as long as the total number of units taken meet the total number required.".
     "</td><td colspan=\"2\" align=\"right\"><b>Grand Total:</b></td>".
```

```
     "<td align=\"right\"><b>".number_format($grandtotalplanned,1)."</b></td>".
     "<td align=\"right\"><b>".number_format($grandtotalrequired,1)."</b></td>".
     "</tr>";
?>
</table></font></center>
</body>
</html>


<!-------------------------------------------------------
- Filename:  showreportcard.php
- Written by: Darryl Scroggins
- Last Modified : 2004-10-25 23:25
-------------------------------------------------------->
<?
header("Location: reportcard.php?studentid=$studentid&gradelevel=$gradelevel
       &semester=$semester&printfmt=1");
?>


<!-------------------------------------------------------
- Filename:  showtranscript.php
- Written by: Darryl Scroggins
- Last Modified : 2005-05-31 00:32
-------------------------------------------------------->
<html>
<head>
    <? include("header.inc"); ?>
</head>

<body>
<?
if (!$studentid) {
   echo "You must select a student<br>";
   die("Click the Back Button and try again.");
}

require_once('dbconnect.php');
$connect = dbconnect();

$query = "SELECT firstname, lastname FROM students WHERE ".
       "studentid=\"$studentid\"";
$result = mysql_query($query) or die("Bad query: $query");
$r = mysql_fetch_array($result);
$studentname=$r["firstname"]." ".$r["lastname"];

$query = "SELECT grade, points FROM gradepoints ORDER BY points DESC";
$result = mysql_query($query) or die("Bad query: $query");
while ($r = mysql_fetch_array($result)){
    $grades[] = $r["grade"];
    $points[$r["grade"]] = $r["points"];
}

$q="SELECT sum(sickdays) as totalsickdays, sum(vacationdays) as ".
   "totalvacationdays, sum(schooldays) as totalschooldays ".
   "FROM monthlyeval WHERE studentid=$studentid";
$result = mysql_query($q) or die(mysql_error());
$nrows=mysql_num_rows($result);
if ($nrows) {
   $r = mysql_fetch_assoc($result);
   $totalsickdays= $r["totalsickdays"];
   $totalvacationdays= $r["totalvacationdays"];
   $totalschooldays= $r["totalschooldays"];
} else {
   $totalsickdays= 0; $totalvacationdays= 0; $totalschooldays= 0;
}

// Tab is used for spacing and to make underlines
$tab="       ";

?>

<center>
<table><tr><td>
<?
echo "<center><h2><i>Dikaios Christian Academy</i></h2>".
     "<h3>Transcript for: <u>  $studentname  </u>   ".
```

```
     "</u></h3></center>\n";

echo "<center>Total Attendance<br>";
echo "Sick Days: <u>  $totalsickdays  </u>  \n";
echo "Vacation Days: <u>  $totalvacationdays  </u>  \n";
echo "Days in School: <u>  $totalschooldays  </u><br><br>\n";


$query = "SELECT subjectid, subject, unitsrequired FROM subjects ".
    "WHERE unitsrequired > 0.0 ORDER BY printorder";
$result = mysql_query($query) or die("Bad query: $query");
echo "<center>Credit Summary<br>\n";
echo "<table border=\"1\"><tr>\n".
     "<th>Subject Area</th>\n".
     "<th>Required</th>\n".
     "<th>Completed</th>\n".
     "<th>Deficient</th>\n".
     "</tr>\n";
while ($r = mysql_fetch_array($result)){
    $sid = $r["subjectid"];
    $subject = $r["subject"];
    $unitsrequired = $r["unitsrequired"];
    $query2 = "SELECT SUM(unitscompleted) as completed FROM courseofstudy ".
        "WHERE studentid=\"$studentid\" AND subjectid=\"$sid\" GROUP BY subjectid";
    $result2 = mysql_query($query2) or die("Bad query: $query2");
    $r2 = mysql_fetch_array($result2);
    $completed = $r2["completed"];
    $totalrequired += $unitsrequired;
    $totalcompleted += $completed;
    $deficient = 0;
    if ($completed<$unitsrequired) { $deficient=$unitsrequired-$completed; }
    echo "<tr>\n".
       "<td>".$subject."</td>\n".
       "<td align=\"right\">".number_format($unitsrequired,1)."</td>\n".
       "<td align=\"right\">".number_format($completed,1)."</td>\n".
       "<td align=\"right\">".number_format($deficient,1)."</td>\n".
    "</tr>\n";
}
    $totaldeficient = 0;
    if ($totalcompleted<$totalrequired) {
       $totaldeficient=$totalrequired-$totalcompleted;
    }
    echo "<tr>\n".
       "<td>Total Credits</td>\n".
       "<td align=\"right\">".number_format($totalrequired,1)."</td>\n".
       "<td align=\"right\">".number_format($totalcompleted,1)."</td>\n".
       "<td align=\"right\">".number_format($totaldeficient,1)."</td>\n".
    "</tr></table></center><br>\n";

$query = "SELECT * FROM courseofstudy cos, classes c, subjects s ".
    "WHERE studentid=\"$studentid\" ".
    "AND cos.classid=c.classid AND cos.subjectid=s.subjectid ".
    "ORDER BY gradelevel,semester";

$result = mysql_query($query) or die("Bad query: $query");

$query2 = "SELECT activities,fieldtrips,note1,note2,note3,note4 ".
    " FROM monthlyeval WHERE studentid=\"$studentid\" ";
$result2 = mysql_query($query2) or die("Bad query: $query");

$classcount=mysql_num_rows($result);

echo "<br>Grade Summary<br>\n";
if ($classcount>0) {
echo "<table width=\"100%\" align=\"center\" border=\"1\">\n".
    "<th>Subject Area</td>\n".
    "<th>Class\n".
    "<th>Hours\n".
    "<th>Units<br>Attempted\n".
    "<th>Units<br>Completed\n".
    "<th>Grade\n".
    "</tr>\n";
    $totalunits=0; $gpa=0;
    while ($r = mysql_fetch_array($result)) {
        $cosid=$r['cosid'];
        $subject=$r['subject'];
        $classname=$r['classname'];
```

```
        $units=$r['units'];
        $hours=$r['hours'];
        $classgrade=$r['classgrade'];
        $unitscompleted=$r['unitscompleted'];
        $totalunits+=$unitscompleted;
        $gpa+=$unitscompleted * $points[$classgrade];
        echo "<tr>".
            "<td align=\"right\" valign=\"top\">".
            $subject.
            "</td><td valign=\"top\">".
            $classname.
            "</td><td valign=\"top\" align=\"right\">".
            $hours.
            "</td><td valign=\"top\" align=\"right\">".
            $units." ".
            "</td><td valign=\"top\" align=\"right\">".
            $unitscompleted." ".
            "</td><td valign=\"top\" align=\"left\">".
            " ".$classgrade.
            "</td>".
            "</tr>\n";
    }
    if ($totalunits>0) { $gpa /= $totalunits; } else { $gpa = 0; }
    echo "<tr><td colspan=\"4\" align=\"right\" >Totals: </td>".
         "<td align=\"center\">".number_format($totalunits,2).
        "</td><td align=\"center\">".number_format($gpa,2)."</td></tr>";

    echo "<tr><td colspan=\"6\">".
         "<center><b>EXTRA-CURRICULAR ACTIVITIES</b></center></td></tr>".
         "<tr><td colspan=\"6\">";
    $extra="";
    while ($r2 = mysql_fetch_array($result2)) {
       $act=$r2['activity'];
       $trips=$r2['fieldtrips'];
       $n1=$r2['note1'];
       $n2=$r2['note2'];
       $n3=$r2['note3'];
       $n4=$r2['note4'];
       if ($act) { $extra.= $act."<br>"; }
       if ($trips) { $extra.= $trips."<br>"; }
       if ($n1) { $extra.= $n1."<br>"; }
       if ($n2) { $extra.= $n2."<br>"; }
       if ($n3) { $extra.= $n3."<br>"; }
       if ($n4) { $extra.= $n4."<br>"; }
    }
    if ($extra) {
        echo $extra;
    } else {
        echo "None<br>";
    }
    echo "</td></tr>";
    echo "</table>\n";
    echo "<br>Teacher's Signature: <u>".$tab.$tab.$tab.$tab.$tab.$tab.$tab.
         "</u> Date: <u>".$tab.$tab.$tab."</u><br>";
    echo "<br>ISP Admin's Signature: <u>".$tab.$tab.$tab.$tab.$tab.$tab.$tab.
         "</u> Date: <u>".$tab.$tab.$tab."</u><br>";
}
?>
</td></tr></table></center>
</body>
</html>


<!-------------------------------------------------------
- Filename:  valid_user.php
- Written by: Darryl Scroggins
- Last Modified : 2005-04-27 20:52
-------------------------------------------------------->
<?
    session_start();
    if (!isset($_SESSION['valid_user'])) {
        header("Location: index.php");
        die();
    }
?>
```

REFERENCES

1. Apache http://www.apache.org
2. DBDesigner http:/www.fabforce.net/dbdesigner4
3. Dikaios Christian Educators Association Handbook, pp. 1-79, May 2002
4. Håkon Wium Lie and Bert Bos, "Cascading Style Sheets", Second Edition, Addison-Wesley, 1999
5. J. Castagnetto, H. Rawat, S. Schumann, C. Scollo, D. Veliath "Professional PHP Programming", Wrox Press, 1999
6. Julie C. Meloni, "PHP Essentials", Prima Publishing, 2000
7. K12Admin A school administration program http://www.k12admin.org
8. Luke Wellington and Laura Thomson, "PHP and MySQL Web Development", Second Edition, Sams Publishing, 2003
9. Mambo http://www.mambo.com/
10. Moodle http://moodle.org
11. MySql http://www.mysql.org
12. Peter Rob and Carlos Coronel, "Database Systems", Third Edition, Course Technology, 1997
13. PDF-PHP http://sourceforge.net/projects/pdf-php
14. PHP http://www.php.net
15. phpMyAdmin http://phpMyadmin.net