Manual

ETM Advising System

Full System
Instructions

April **2019**

The Instruction Guide

This Guide has been created to provide answers to all questions on how to use the system and how-to setup all necessary components.

The Engineering Technology & Management System has been created to help innovate the process of advising students. Features include a fully functioning DARS Report stripper. A login system connected to a MySQL Database.

Technologies Used:

MySQL ~ Database

MySQL is a highly regarded database system that is widely used and has a free community edition

Node.Js ~ Web Server

Node.Js is a widely used language to implement web applications. Its powerful Javascript runtime allows for high efficiency and accuracy. Allowing tons of requests to be made at the same time. It is highly scalable and regards HTTP first class.

Python ~ Parser Language

Python is one of the top scripting languages out there. With a wide library and a huge community, the programming language is chosen to easily support the stripping of the DARS reports.

Database Installation

1. First Install 'MySQL Community Edition' to the computer that will host the system.
2. Make sure to install MySQL Server and MySQL Workbench
   a. For Windows you can install the MySQL Installer
3. Once you have MySQL Workbench installed and open click the plus button next to MySQL connections. This will open the windows shown in Figure 1. Here you can name a new connection, setup the desired IP Address. For the computer to use its local connection keep the hostname 127.0.0.1. Keep the port number 3306. Choose a username and click 'Store in Vault..' to setup a password for your database connection.
4. Once your connection has been created it should now be available under the MySQL Connections. Double click on it and sign in with your password.
5. On the next page you will see the MySQL Workbench main page. Click on Server -> Data Import
   a. From the included files import the file 'SQL -> etmdatabasesettings.sql'
6. You should now see your database schema in the left panel called etm (do not change its name)
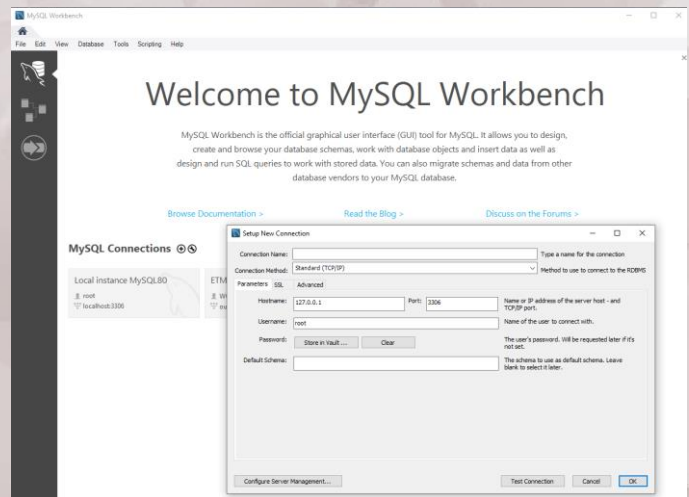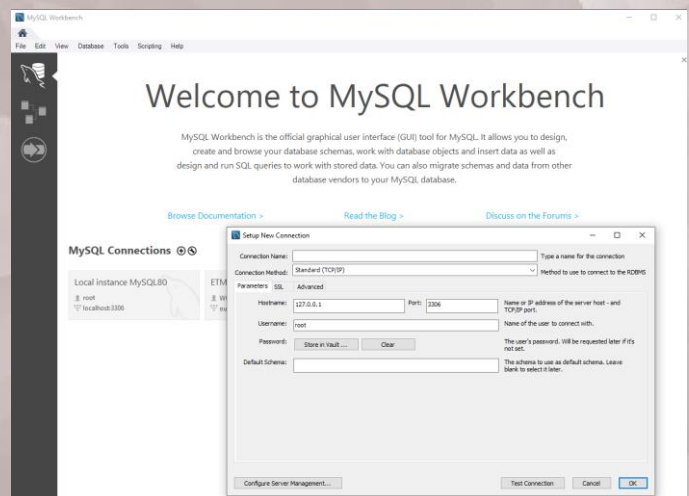7. Your database is now running on this machine under the specified IP Address at port 3306.

Figure 1

Figure 2

DARS PARSER Installation

1. **First Install Python 2.7 and Pip**
   a. https://www.python.org/downloads/
2. **Then run these commands**
   a. **pip install pipenv**
   b. **pipenv install re**
   c. **pipenv install csv**
   d. **pipenv install urllib**
   e. **pipenv install pathname2url**
   f. **pipenv install collections**
   g. **pipenv install webbrowser**
   h. **pipenv install glob**
   i. **pipenv install mysql-connector-python**
   j. **pipenv install pdfminer**
   k. **pipenv install io**
   l. **pipenv install tabula**
   m. **pipenv install pandas**

Web Server Installation

1. **Download Node.js onto the computer that will host the application**
   a. **https://nodejs.org/en/download**
2. **Choose the Windows or Mac Installer**
3. **Once Installed you are all set to go**
4. **To run your application, open a terminal window that is pointing inside the folder of the project files**
5. **Now run the command 'node setup.js' or 'forever server.js' (without quotations)**
   a. **If for any reason the application breaks. Quit the terminal window running the program and re-run this command**
6. **Now your website is up and running.**
7. **Next let's setup the database settings**
8. **The Program runs on the host machine using the IP Address of the host machine. It is up to the user to figure out how to route the traffic to a URL**
9. **The program is setup to use port 80 which is the default webpage port. So that access does not require a port number.**

Figure 3
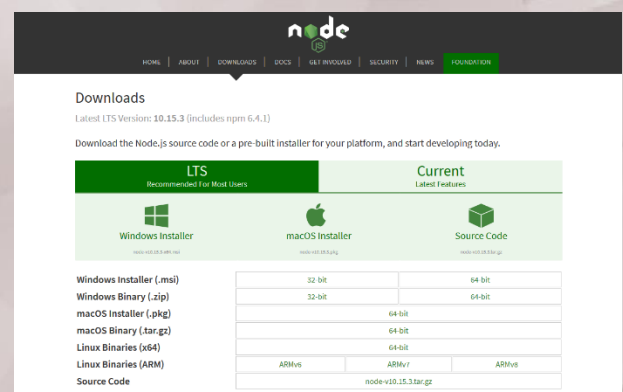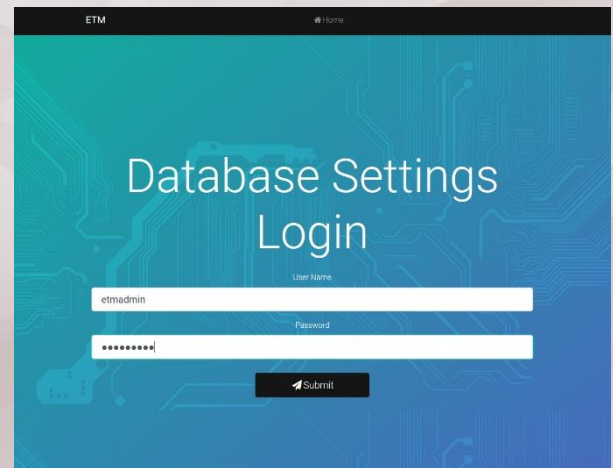
Figure 4
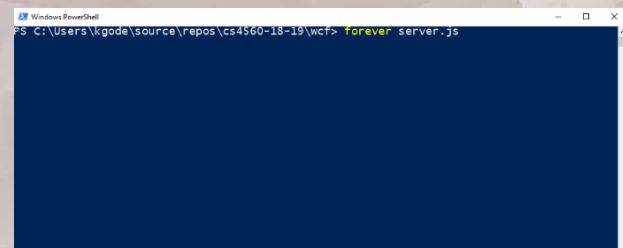
Web Server Database Settings

1. Once your application is running on your host machine.
2. Go to 'http://<IP Address or Hostname>/serve
3. The Login Credentials are predefined for non-database access
   a. Username: etmadmin
   b. Password: Password!
4. Once here enter the credentials that you used to setup your database
   a. Host Url: IP Address or Host URL
   b. User: MySQL Username
   c. Database Name: etm
      i. This MUST be 'etm'
   d. Database Password: MySQL database password
5. Your Database should now be hooked up to the web application and you may now register accounts and login.

Figure 5
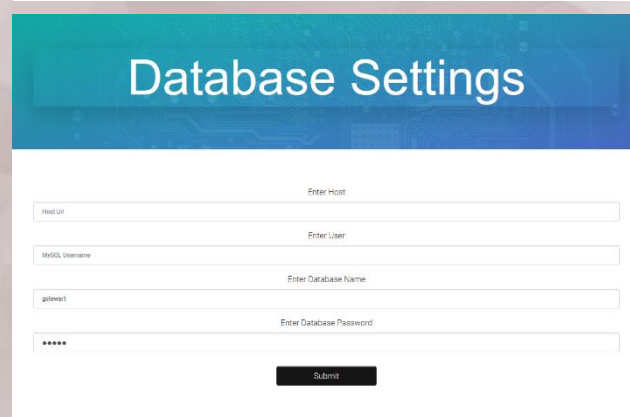
Figure 6

Account Creation & Login

1. **The Homepage displays two buttons. One for login and one for registration**
2. **Users are matched to their students based on their Ohio Emails displayed on DARS reports.**
3. **When Registering a user is required to enter an Ohio email address.**
4. **Once you hit submit the users password is highly encrypted and stored inside the database inside the 'accounts' table. As shown in <u>Figure 8</u>.**
5. **To create administration accounts first register an account as a regular advisor.**
6. **Then go into the database right click on accounts and click 'Select Rows – Limit 1000'**
7. **Find the user you wish to make admin. Go to the column labeled 'type'**
8. **Change the user from 'atype' to 'admin'**
9. **As shown in <u>Figure 9</u>.**

Figure 7

Figure 8

Figure 9

Main Page - Student Catalog

1. <u>Figure 10</u> shows the main page for an advisor. Here you can view all the students connected to that advisor's email they used to register with.
2. Along the navigation bar we have
   a. Add Student: Upload DARS
   b. Course Catalog: View Estimated Students
   c. Manual: View user manual
   d. Change Password
   e. Logout
3. Below the navigation bar we display all available students for an advisor.
   a. Here you can click on
      i. Student Progress: DARS Report
      ii. Student Plan: 4-year plan
      iii. Archives: Previously saved archives for that student
4. <u>Figure 11</u> shows the main page for an administrator
5. Along the navigation bar we can
   a. Students: View all students in the database
      i. Similar view as <u>Figure 10</u>. Administration can also delete students.
   b. Edit Layout: Control the layout for all users
   c. Manual: View user manual
6. The user accounts are all displayed on the administration main page. You can delete/reset any accounts password.

Figure 10

Figure 11

Student Program Of Study

1. <u>Figure 12</u> shows the whole program of study page.
2. Under 'ETM Bachelor of Science' you can see the auto generated GPA, Student information, change the advisor using the advisor's email, and use 'Catalog Number' to change the catalog year.
3. Below you can see six tables with different headings. Each heading is matched to a course to make it easier to read and more quickly find courses*.
4. Within each table, you can change grades, semesters, and course codes. Course codes match to courses in the database and will automatically display course name, hours, tier, and prerequisites
5. Double Clicking on an empty box will show all available options in the database
6. Advanced Standing will turn green when checked and save to the database
7. Two buttons are fixed to the bottom of the screen as you scroll.
   a. Download (Left) will download all the student's current shown data to a CSV file and can be reached by looking at the archives on the home page.
   b. Submit (Right) will submit all changes to the database**.

*Note : When adding a course, make sure the heading matches the table you would like it to be in.
**Note : These changes will affect other pages such as the Program Planner.

Figure 12

Program Of Study Layout

1. **When a new catalog is made or the layout needs updated, the Program of Study Layout page should be used. This allows admin uses to rename and rearrange any course, table, or placeholder.**

2. **Figure 13 shows the page reached when you click 'Edit Layout' (only admin users have this button).**

3. **Figure 14 shows the Catalog Course Layout page. Here you can edit the ordering of courses and which table they are placed inside. Course placement for each table starts at 1*.**

4. **Figure 15 shows the Placeholder Layout Page. This is set up the same as the course layout, but is for the placeholders used in each position before they have a course.**

5. **Figure 16 shows the Heading Layout Page. This is a way to organize the table name and order on the POS page.**

*Note : Putting a 0 disables a course, heading, or placeholder.

Figure 13

Figure 14

Figure 15

Figure 16

Additions & Deletions

1. <u>Figure 17</u> Shows the course addition page (The Placeholder addition is similar but less fields are required). Here you can add a course to the database. Most fields are required, and the formatting is enforced so that everything is compatible with the DARS reports. All available headings in the database are also listed in the 'Choose a Catalog Heading' option.

2. <u>Figure 18</u> shows the heading addition page. Here you can add new headings by catalog year, you can choose a name, and then you choose a table in the program of study for that heading to be in.

3. To disable headings, use position 0.

4. Deletion of a course will remove it from the database which includes removing the course from student data.

5. You can update information for a course by using the 'Course Add' page and using the same course code and catalog year.

Figure 17

Adding a Course by the same Code and Catalog will update other values.

Enter Course Code

Should Match DARS. Example: ETM1010

Enter Course Name

Course Name

Enter Course Hours

Course Hours

Enter Course Tier

Course Tier

Enter Course Prereq's

Course Prereqs

Enter Course Catalog Year

Example: 2015-16

Choose a Catalog Heading

Business Courses

Submit

Figure 18

To Disable Heading Use Position 0

Enter a Catalog Year

Example: 2015-16

Enter Heading Name

Ex: Free Electives

Choose a table

Submit

DARS Parser

1. <u>Figure 19</u> shows the 'Add Student' Page where you can upload DARS PDF's and they will be run on the python script and put into the database.
2. Inside the DARS Parser python file, make sure the database settings are changed to reflect your local database settings. Otherwise the parser will be unable to submit its findings to the database.

Figure 19
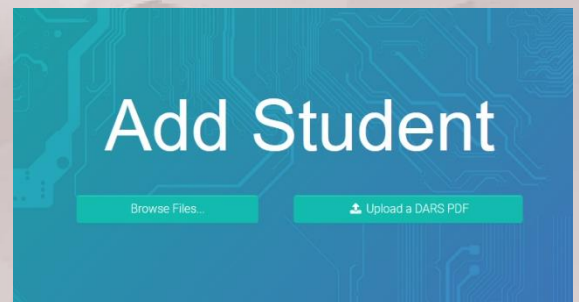
Code Catalog – Page 1

Folder : Dars
1. **DP2.py : DARS Parser (Contains code to parse DARS PDFs)**
   a. **Requires Python & Python Dependencies listed in file**
2. **DAR's files are imported into the 'Dars' Folder and then run with DP2.py**

Folder: node_modules
1. **This Folder contains code installed through the Node.js package manager called 'npm'**
2. **If for any reason the program cannot run, you can try deleting this entire folder and then running the command 'npm install' inside the project folder**

Folder: routes
1. **This folder contains all the Node.js files that power the web server.**
   a. **File: about.js**
      i. **This file is used to serve the About Page**
   b. **File: addstudent.js**
      i. **This file is used to serve the Add Student Page for the advisor view**
   c. **File: adminchangepassword.js**
      i. **This file is used for password changing with all accounts through the administration account.**
   d. **File: admincourseedit.js**
      i. **This file is used to serve the Program of Study Layout Pages**
   e. **File: administration.js**
      i. **This file is used to serve the Admin Home Page upon login**
   f. **File: archives.js**
      i. **This file is used to serve the Archives page for the advisor view**

Code Catalog – Page 2

g. File: archives2.js
   i. This file is used to serve the Archives Page for the admin view
h. File: auth.js
   i. This file is used to define the authentication enforcement for all pages. This is part of the 'Passport' NPM package
i. File: changepassword.js
   i. This file is used to serve the change password for the main page (before login)
j. File: courseedits.js
   i. This file has been deprecated but kept for future implementation
k. File: courseloadingview.js
   i. This file is used to serve ALL course loading view's
l. File: db.js
   i. This file is used to define the database settings for the web server
m. File: db2.js
   i. This file is also used to define the database settings but enables MySQL multiple statements for the Program Of Study pages.
n. File: deletestudent.js
   i. This file is used to serve the delete student operations in student catalog admin view
o. File: deletion.js
   i. This file is used to delete accounts in the admin home page view
p. File: download.js
   i. This file is used to serve the archive download operation for the advisor view

Code Catalog – Page 3

q. File: downlaod2.js
    i. This file is used to serve the archive download operation for the admin view
r. File: errorpage_admin.js
    i. This file is used to serve the error pages for the admin view
s. File: errorpage.js
    i. This file is used to serve all of advisor view and basic page view error pages
t. File: houme.js
    i. This file is used to serve ALL of the home pages for both views
u. File: index.js
    i. This file is used to serve the main page when you access the website
v. File: info.js
    i. This file is used to serve the information pages scattered throughout the web pages
w. File: license.js
    i. This file is used to serve the License web page
x. File: loading.js
    i. This file is used to serve the loading pages for the web server
y. File: login.js
    i. This file is used to serve the login operation
z. File: logout.js
    i. This file is used to serve the logout operation
aa. File: passport.js
    i. This file is used to serve the login operations. This file checks for the correct password and authenticates users

Code Catalog – Page 4

---

bb. File: privacypolicy.js
- i. This file serves the privacy policy page

cc. File: programofstudy_admin.js
- i. This file serves the program of study view for the admin accounts

dd. File: programofstudy.js
- i. This file serves the program of study view for the advisor accounts

ee. File: programplanner_admin.js
- i. This file serves the program planner for admin accounts

ff. File: programplanner.js
- i. This file serves the program planner for advisor accounts

gg. File: register.js
- i. This file serves the registration operations

hh. File: submit.js
- i. This file serves the Thank Page for the Program Of Study

Folder: SQL
1. File: etmdatabasesettings.sql
- ii. This file is used to setup your Database by importing this file into your schema through the MySQL Workbench

Folder: test
1. This folder contains files used to test the program to make sure there were not issues.

Folder: TheArchives
1. This folder is used with the program to store all of the Program Of Study Submissions

Code Catalog – Page 5

Folder: views
1. **This Folder is used to store all the files that are used to render every single page. It also includes all the CSS and SASS files.**
2. **Folder: assets**
   a. **This folder contains assets used within the program**
3. **Folder: js**
   a. **This folder contains javascript files used within the HTML**
4. **File: add*.pug**
   a. **These files are used to display pages that allow a user to add something to the program**
5. **File: admin*.pug**
   a. **These files are used to display admin views**
6. **File: archive*.pug**
   a. **These files are used to display the archive pages**
7. **File: change*.pug**
   a. **These files are used to display the change password pages**
8. **File: courseloadingview*.pug**
   a. **These files are used to display the Project Students for ETM courses**
9. **File: edit*.pug**
   a. **These files are used to add and change the Layout of the POS**
10. **File: error*.pug**
    a. **These files are used to display error pages.**
11. **File: houme*.pug**
    a. **These files are used to display the home pages**

Code Catalog – Page 6

12. File: index.pug
    a. Main page
13. File: info*.pug
    a. All pages associated with the information pages
14. File: *.pug
    a. Pug Files are JavaScript Template Pages that are rendered into HTML when requested for

File: app.js
1. This file is the powerhouse of the whole entire webserver. Without it the program would be nothing. Inside we have HTTP security features, user sessions, routes, and catch all un-caught errors.
2. The Routes define how pages are accessed through URL's

File: dbcreate.py
1. This file is used to change the database settings for the programs so that any user can connect to whatever database they set up without having to access any of the code

File: package-lock.json
1. This file is auto created when you run 'npm install'. It is not recommended that any changes are made to this file.

File: package.json
1. This file is auto created when installing new NPM packages. This file lists all the NPM package dependencies and it is not recommended that you make any changes to it.
2. NPM is a Node.js package manager which allows you to install many packages from the NPM community.

File: server.js
1. This file is used to run the server.

Code Catalog – Page 7

File: setserver.js
1. **This file is used to serve the database settings page** http://<website>/serve
2. **This allows you to update the database settings as defined early in this user guide.**

File: setup.js
1. **This file is the main file which creates the listener on port 80 to allow communication between users and the webserver. This file should never have to change.**