<u>The Vulcan</u>

<u>Schedule Planner</u>

Instructor Comments/Evaluation

Table of Contents

Introduction

When it comes to scheduling, on the surface it appears to be easy, however creating a schedule and even knowing what classes you need to take and when can all be a big challenge. This challenge is accentuated with incoming students and freshmen. Students who are coming in also need to worry about how their SAT/ACT scores and any placement tests they have taken with Cal U. This can cause a stressful situation for the student because if he/she does not do well on these tests their schedule could be intimidating and quite demanding. Especially if the student feels like it will take them more than four years to graduate. If a student does not do well in mathematics most four year plans have students start in calculus 1 and if a student does not meet the requirements to start there is a feeling of being left behind.

With our program The Vulcan Schedule Planner we hope to bring clarity to the students of their schedule and what to expect in their major. Our schedule planner will give students the ability to generate a schedule that will go through their major all the way up until they graduate. The generated schedule will provide a four year grad plan to the best of its abilities. The Vulcan Schedule Planner will also take into account spring and fall classes, so if a student would join in spring their generated schedule will reflect that. Students will also be able to manually put in the classes they have taken to generate a schedule for a sophomore or a junior. There is also the ability to save the generated schedule by downloading a file and when you want to bring up that schedule with ease we have an uploader that will take the info on the file and instantly have every class taken recorded.

Project Overview and Application:

Motivation:

Our motivation to create this project began from frustrations that came from the scheduling process, and the complexity of V.I.P. and its mountains of information that can be difficult to sort through. It was common for us and other students that we have spoken with to voice their frustrations with trying to understand where this information was, and make sense of the Degree Works degree layout worksheets. From there it can be difficult to get information on the classes you need, and see what you need to take next. We have decided to mimic their layout of courses on the main degree worksheet, as that is most likely what the user is accommodated with.

Because of these roadblocks and frustrations we decided we wanted to try and make something more user friendly. We wanted to be able to convey the information users are typically looking for in many less clicks than what is required when using V.I.P. and Degree Works. We wanted to make it a website that does not require a login so that it is quick and easy to access for any type of student or faculty member.

Comparison to Existing Similar Products:

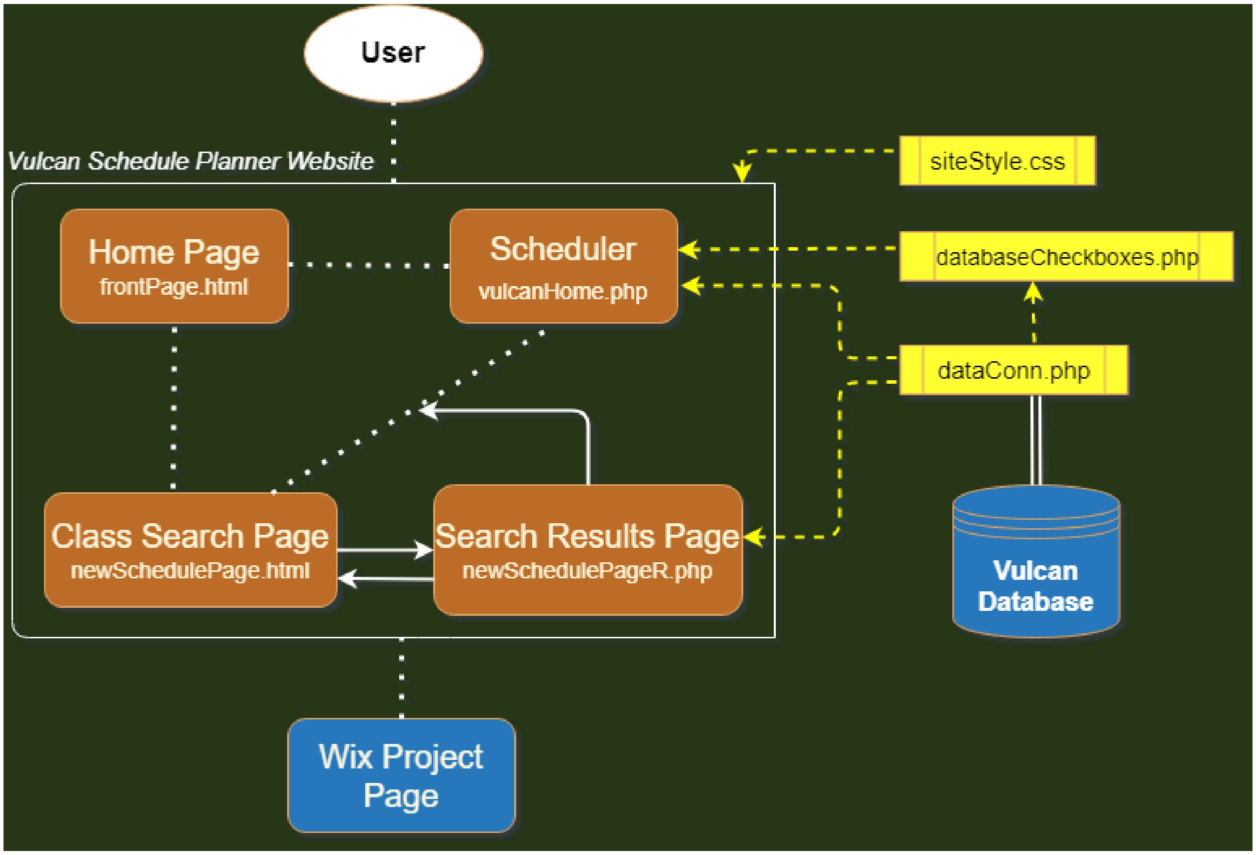
There are not many similar products offered to Students at Cal U. The closest thing that would be similar to this is Degree Works, which is accessible through V.I.P. Our website is easier to access since we don’t require a login, and the functions are not spread apart across different sections of V.I.P. There is scheduler functionality in Degree Works, but it is not easy to find and use. Nobody in our group knew it existed before hearing about it from a faculty member when we started working on the project. It does not provide clear instructions on how to use it, and can be overwhelming if the student is just a freshman. Our website is intuitive and the only input required in most cases would be the user clicking on some checkboxes. Other than using Degree Works the student would have to just schedule a time to see their academic advisor, and go get in person guidance on their scheduling. Without system, these meetings could possibly not even need to occur, or be much quicker than normal. This will help free up time for the advisor to help more students. Because of these reasons we believe that our system will be superior to the user.

Community & Social Implications of our Project

The Vulcan Schedule Planner will appeal to both new and returning students of California University of PA, making the course scheduling process easier. If this project were to be improved and expanded upon, it could be used by more of the student body, as well as the Scheduling Center staff. As it is, the project primarily helps Computer Science students plan their schedule for their time at Cal U. This gives the CSC major an unique interest to new or transfer students. It will be easier for them to plan what classes to take, when to take them, and see the overall plan of their courses. This means that it will be easier for them to graduate, saving students time and money in the long run.

There can be no drawn conclusions about the impacts of this project on the community of the California PA.

System Block Diagram



The system block diagram above represents how the various components of the vulcan scheduler interact. The brown boxes represent the different webpages of the site and the file that creates them. Navigation between these pages are represented by the white dotted lines and solid arrows. Through a global web navigation methodology, a user is able to navigate from any page to any other page except for the search results page. From the class search page, a user can navigate to the results page and enter back into the main navigation structure from there. Also any page on the website can navigate to the Wix project page through the use of the navigation bar.

The yellow blocks represent the utility files that are used by the website files. The yellow dashed arrows represent a file calling a utility file. Every web page file uses the *siteStyle.css* file for graphical styling and formatting of the different webpages. While the scheduler page, *databaseCheckboxes.php*, and search results page use *dataConn.php* to connect to the vulcan database that holds class information. Also the scheduler page calls *databaseCheckboxes.php* to create all the toggleable class checkboxes.

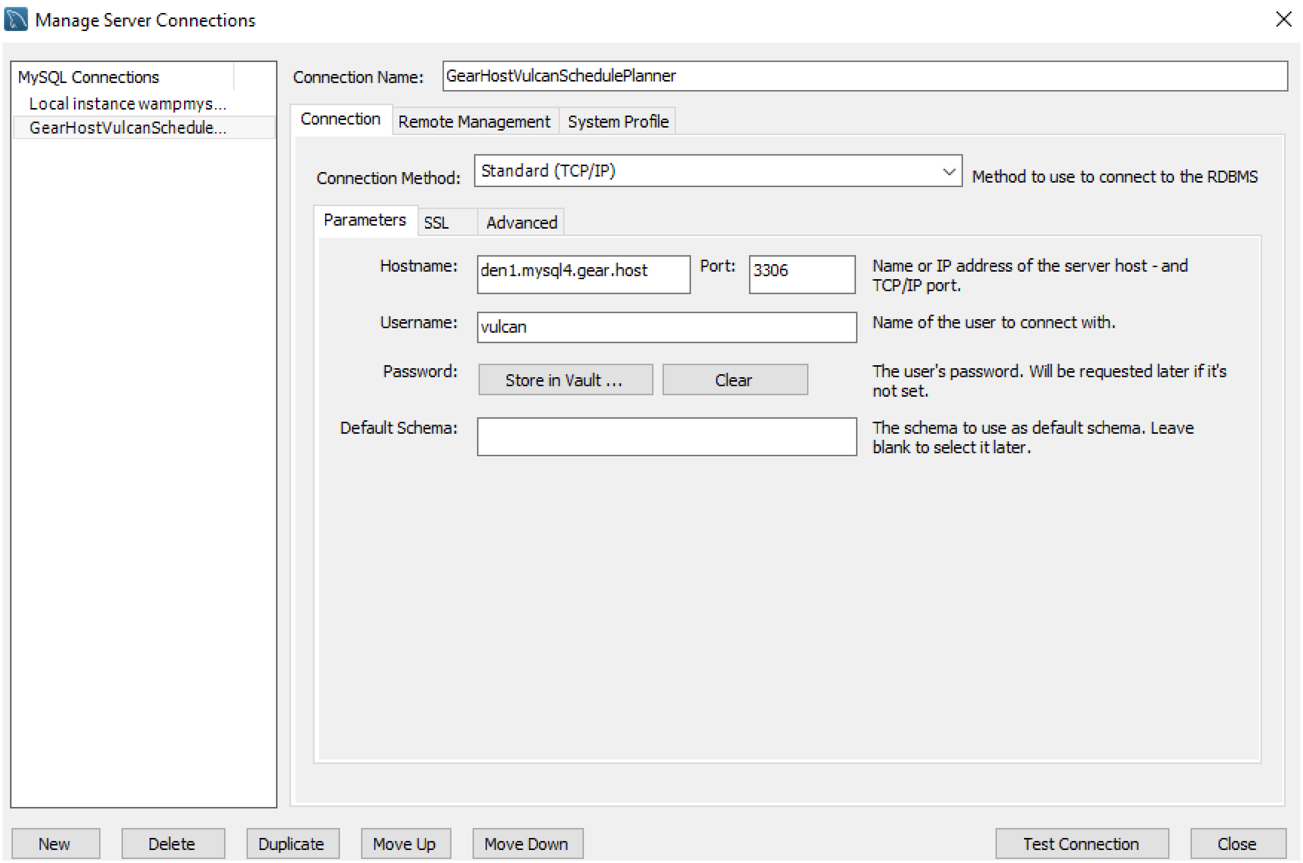
Description

The Vulcan Schedule Planner is a program that is run through a website to develop a schedule for their 4 years of college. This primarily is to help students, incoming or already in college, to help plan out their future in college. The schduleplanner will can make a schedule from scratch or use inputs from the user to generate an optimal schedule for the student. This schedule also takes into account whether the student is in the spring semester or fall semester of college. Incoming students can use their SAT and ACT to affect how their schedule is made. A student already in college can check off what classes they have taken or are in to get what their remaining schedule will look like. Once you are done with the scheduleplaner a person can download a text document that you can have uploaded from any computer to bring up a previous schedule.

Administrator

To make changes to the database, administrators will download and install wampserver64 as well as mysql workbench. We recommend downloading wampserver from https://sourceforge.net/projects/wampserver/ and mysql workbench from https://downloads.mysql.com/archives/workbench/ . Wampserver allows you to change and edit all of the database using PHPMyAdmin. Mysql workbench allows you to import and export the data to the database.

First you must download the scheduleplanner database from gearhost. To do this first open MySQL Workbench and in the center click the '+' next to MySQL Connections. Fill in the following information to the window that opens:


Manage Server Connections

MySQL Connections
Local instance wampmys...
GearHostVulcanSchedule...

Connection Name: GearHostVulcanSchedulePlanner

Connection Method: Standard (TCP/IP) — Method to use to connect to the RDBMS

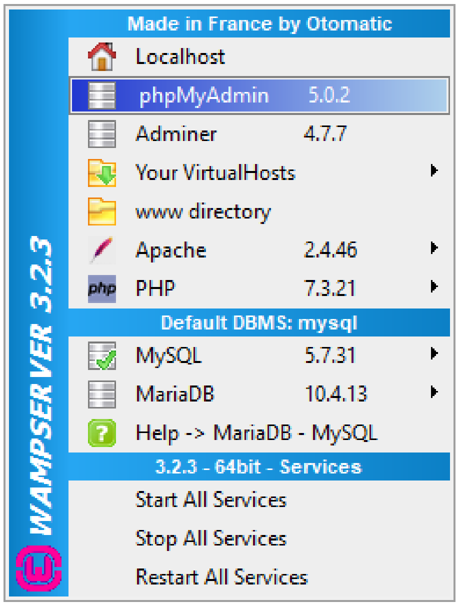
Hostname: den1.mysql4.gear.host — Port: 3306 — Name or IP address of the server host - and TCP/IP port.

Username: vulcan — Name of the user to connect with.

Password: The user's password. Will be requested later if it's not set.

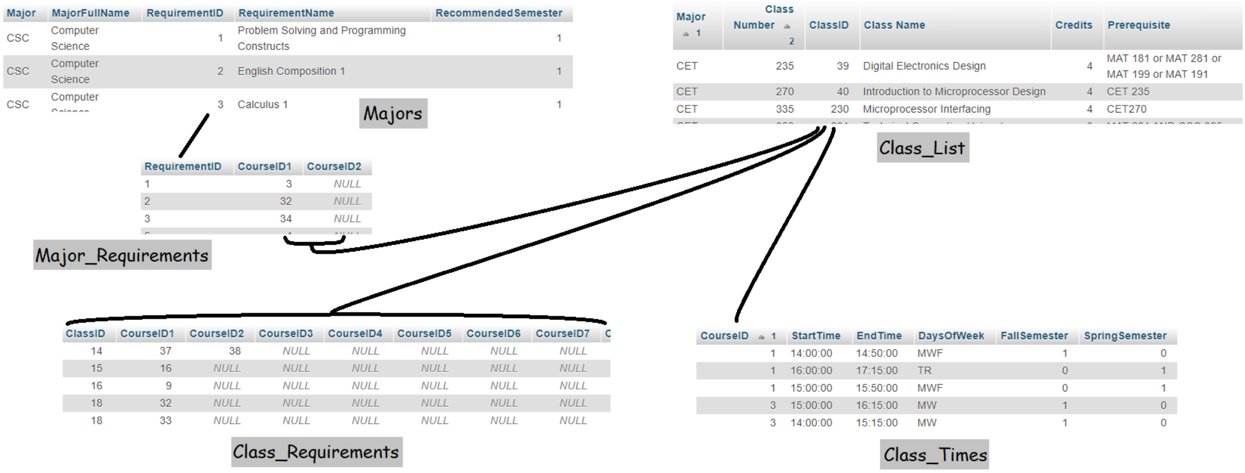
Default Schema: The schema to use as default schema. Leave blank to select it later.

Test the connection to make sure that everything was entered correctly. If a password is prompted use "masterchen$" as it is the master password. Close the window and click the new connection to open it. At the top select "Server" and in the drop down select "data export". In Export Options change it to Export to Self-Contained File and click start export. You might need to resize mySQL Workbench to see the start export button. Once it is exported to your computer make note of where it is saved to and you can now open it in wampserver.



Once you run wampserver you will see the wampserver icon in your system tray on the bottom right. A red symbol represents that it is currently stopped, a yellow symbol represents that it is stopping or starting, and a green symbol means that it is currently running. Single click on the icon once it is green and select “phpmyadmin”. It will take you to a login screen, use username “root” and leave the password blank. If this is your first time doing this create a new database in the left panel and name it “scheduleplanner”. At the top select Import, choose the file that you saved from mySQL Workbench and click “Go” at the bottom right.

In the left panel click on the scheduleplanner database, and click on the table that you want to change. Typically you will only have to change class_times or class_list. Please refer to the figure below to see how the tables are linked. At the top you can add rows using insert, or you can change rows by clicking the SQL tab and using an SQL command. Please note that you should be as specific as you can be with SQL as incorrect statements can delete or alter more than you wanted. I highly recommend pressing the “simulate query” button to see how many rows of the tables will be affected.



Major	MajorFullName	RequirementID	RequirementName	RecommendedSemester
CSC	Computer Science	1	Problem Solving and Programming Constructs	1
CSC	Computer Science	2	English Composition 1	1
CSC	Computer	3	Calculus 1	1

Majors

Major ▲ 1	Class Number ▲ 2	ClassID	Class Name	Credits	Prerequisite
CET	235	39	Digital Electronics Design	4	MAT 181 or MAT 281 or MAT 199 or MAT 191
CET	270	40	Introduction to Microprocessor Design	4	CET 235
CET	335	230	Microprocessor Interfacing	4	CET270

Class_List

RequirementID	CourseID1	CourseID2
1	3	NULL
2	32	NULL
3	34	NULL

Major_Requirements

ClassID	CourseID1	CourseID2	CourseID3	CourseID4	CourseID5	CourseID6	CourseID7	C
14	37	38	NULL	NULL	NULL	NULL	NULL	
15	16	NULL	NULL	NULL	NULL	NULL	NULL	
16	9	NULL	NULL	NULL	NULL	NULL	NULL	
18	32	NULL	NULL	NULL	NULL	NULL	NULL	
18	33	NULL	NULL	NULL	NULL	NULL	NULL	

Class_Requirements

CourseID ▲ 1	StartTime	EndTime	DaysOfWeek	FallSemester	SpringSemester
1	14:00:00	14:50:00	MWF	1	0
1	16:00:00	17:15:00	TR	0	1
1	15:00:00	15:50:00	MWF	0	1
3	15:00:00	16:15:00	MW	1	0
3	14:00:00	15:15:00	MW	1	0

Class_Times

For reference here is an example: Let's assume that you needed to enter new times for CSC 120. I recommend removing all pre-entered times from the table before adding new times. First check what classID CSC 120 corresponds to, in this case it is classID 3. So in the SQL tab of phpmyadmin run the following statement:

```
DELETE FROM `class_times` WHERE `CourseID` = 3
```

That statement will delete all entries in the table class_times that have a courseID/classID of 3. Now that all of the times for CSC 120 have been removed go to the insert tab and enter all of the new times into the system. Please note that this uses military time so a class from 3pm to 3:50 pm should be entered as 150000 for a start time and 155000 as the end time.

Once you are done making changes you must upload it to gearhost for the changes to be finalized. In the top bar of phpmyadmin click export and export the database. Then go back to MySQL Workbench and in the server tab again click data import. Select Import from

Self-contained File and select the file. Start the import and once completed the database should be updated correctly.

Project Implementation Details

Since creating the original design document we have made some notable changes that will be listed here. The largest change made to the project was switching from an application to a web-based scheduler. This allows for easier access to the scheduler and lets people access it from their mobile devices and other computer operating systems instead of just windows computers. This involved changing to using a database rather than having all of the included classes hard coded. This also meant that we would be using a variety of different types of programming including HTML, CSS, javascript, and PHP. Quickly after we began creating the scheduler we made the decision to focus on the computer science major rather than trying to create a scheduler that wouldn't be as refined for all majors at Calu.

Challenges with these changes meant that we would be self learning new programming languages and more. This meant that we had to teach ourselves HTML, CSS, javascript, and PHP in a single semester while handling the workload that our other classes gave us. In addition to just learning all of these from scratch we had to get them to interact with each other, find hosting for both the website and database, and find new ways to let administrators edit the database and system so that maintaining the scheduler can be done after graduation.

Use of Software Engineering Principles

Our team followed the project construction standards and methods discussed in CSC 490 (Senior Project I) in the previous semester. We used these methods to plan out the several different phases of the project. This includes details such as when different stages of the program needed to be done, and who was leading the different parts of development.

The requirements and design phase was mostly completed in senior project 1. In this phase we prepared documentation and planning for what the project would be, what is user experience would be like, and the types of users we would be serving. Much of this changed in the beginning of this Spring semester, when we decided to move to a website and database approach, rather than write a more traditional program. Because of this some of our early documentation does not reflect what we achieved in the final product.

Next was the design phase. This is where we broke down how each part of the project will work. We looked into what information each part of the program will need access to, and what methods we would use to implement the solutions we hoped to achieve. We created diagrams and flowcharts to depict how the information and work will flow from one part to the next.

The final phase was the implementation phase. In this phase we would be implementing the processes and programming techniques we planned out and were going to use. To help us stay on track and monitor our time in this phase we created a Gantt chart. We were able to follow it well. At the end we had a bit of a rush to make sure we could finish. We were able to, and we believe that us not having the extra week that would come from having a Spring break caused this slight crunch at the end. As we completed each module of the design we tested to identify any errors before moving on. Throughout each phase we maintained the programming principles we learned at Cal U. We made sure we were utilizing structured programming to create clear, modular, and readable code.

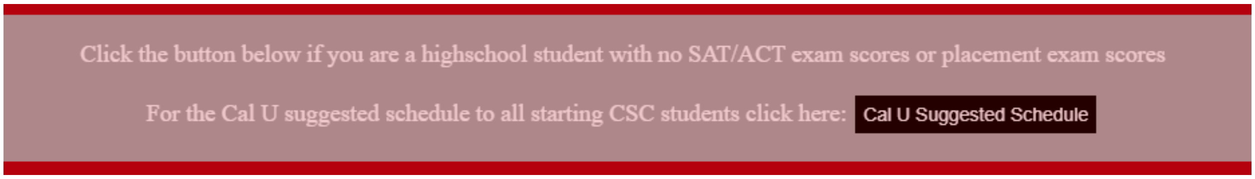
Users & Usages

______To access our project, the user must go to (https://mer3942.wixsite.com/vulcanplanner). This webpage will host all of our documents and tell the user about our project, the team, and some of our motivations. From here you can also get to the page where we have all of our weekly updates. To access the scheduler the user must click the "Start a Schedule" button at the top of the page in the very middle of the screen. That button will take the user to (http://vulcanplanner.gearhostpreview.com/). This site will host all of the webpages we coded ourselves. It is hosted by GearHost, which will contain all of our webpages and the database files.

______There are 3 main types of users that we envision this program will be used by. The first one is an incoming student who is new to Cal U. They could still be in High School or they could be anybody that has never taken courses at Cal U. The second type of user is a student who is currently enrolled at Cal U, and would like to know more about the CSC major and a typical schedule. The needs of this user will be very similar to the needs of a faculty member at Cal U. For this reason we will treat them as the same. The third type of user is an administrator. The administrator's needs were just covered so we will not discuss what they will need as much in this section.

User: New Student & New Student Scheduling:

This group of users will contain students who have not taken any courses before at Cal U. Mainly this will include students who would be looking to transfer into Cal U, and also high school students. Within the group of high school students, there will be two groups. One group is the students who have taken the SAT or ACT before, and know their scores. The other group is potential students who have not taken the SAT or ACT or any placement exams. For the students who have not taken any of these exams, we have a button and display that will link these users to the recommended schedule for all CSC students on the Cal U official site (seen below). This schedule will be somewhat vague, but that is due to use not having any information to go off of.

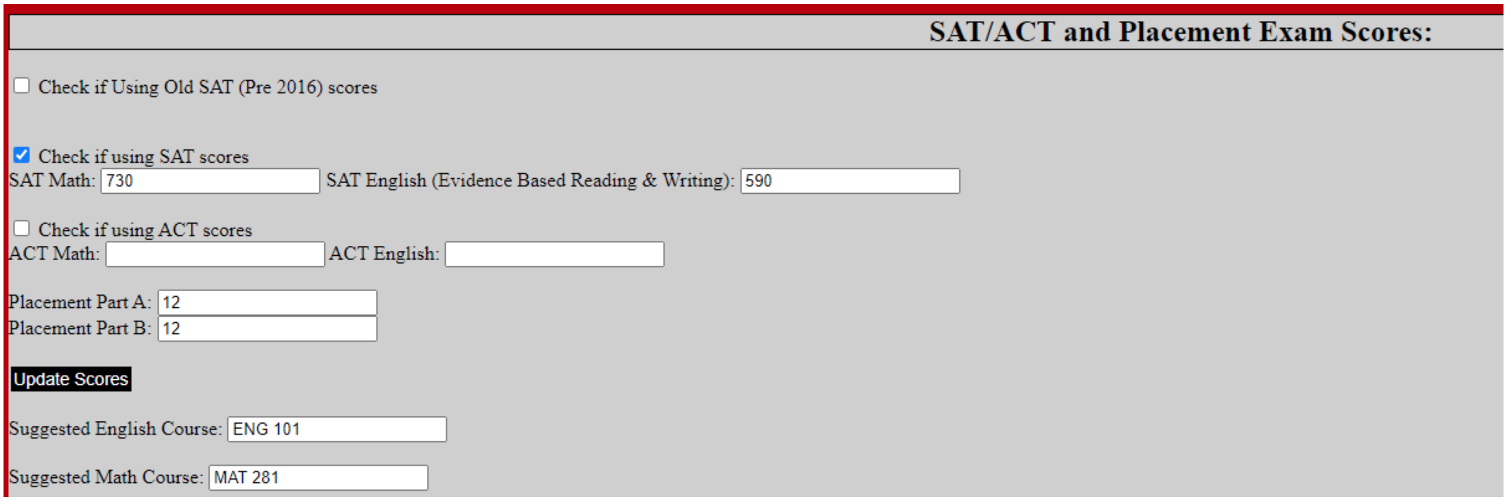


Click the button below if you are a highschool student with no SAT/ACT exam scores or placement exam scores

For the Cal U suggested schedule to all starting CSC students click here:

For students that have taken some of these placement exams and SAT or ACT, we have a section for the user to input these scores. From there the user can click the "Update Scores" button and it will display the recommended MAT and ENG courses for the user to start with. We determine these courses based on the guidelines made available by Cal U at the link (https://www.calu.edu/inside/student-resources/academic-success/testing.aspx). We also have the option for the user to choose the old (pre 2016) SAT or the new SAT exam. The input fields and "Update Scores" button is visible below.

As you can see all the user will have to do is enter their scores for the SAT or ACT and their placement exam scores and they will be given the suggested MAT and ENG starting courses. Once the user has these starting courses they can check off the boxes that are classes



SAT/ACT and Placement Exam Scores:

□ Check if Using Old SAT (Pre 2016) scores

☑ Check if using SAT scores

SAT Math: 730 SAT English (Evidence Based Reading & Writing): 590

□ Check if using ACT scores

ACT Math: ACT English:

Placement Part A: 12

Placement Part B: 12

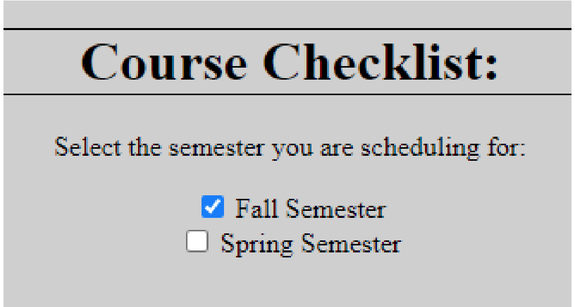
Suggested English Course: ENG 101

Suggested Math Course: MAT 281

they will not have to take. Then when they go to generate their schedule they will start at the recommended courses based on the scores entered.

The user will then need to select which semester they are scheduling for. There are two options, Spring and Fall. Once the user selects one of these boxes, the other one will be unchecked so that both cannot be selected at the same time. This can be seen in the image below.

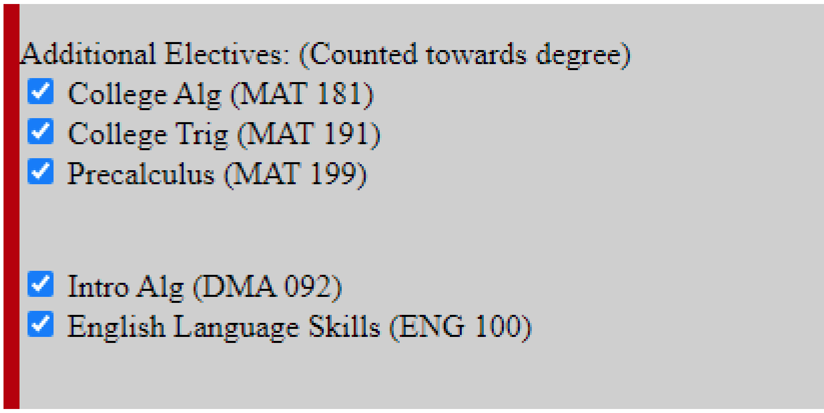
Now the user will select the courses that lead up to the suggested MAT and ENG course.



Course Checklist:

Select the semester you are scheduling for:

☑ Fall Semester
□ Spring Semester

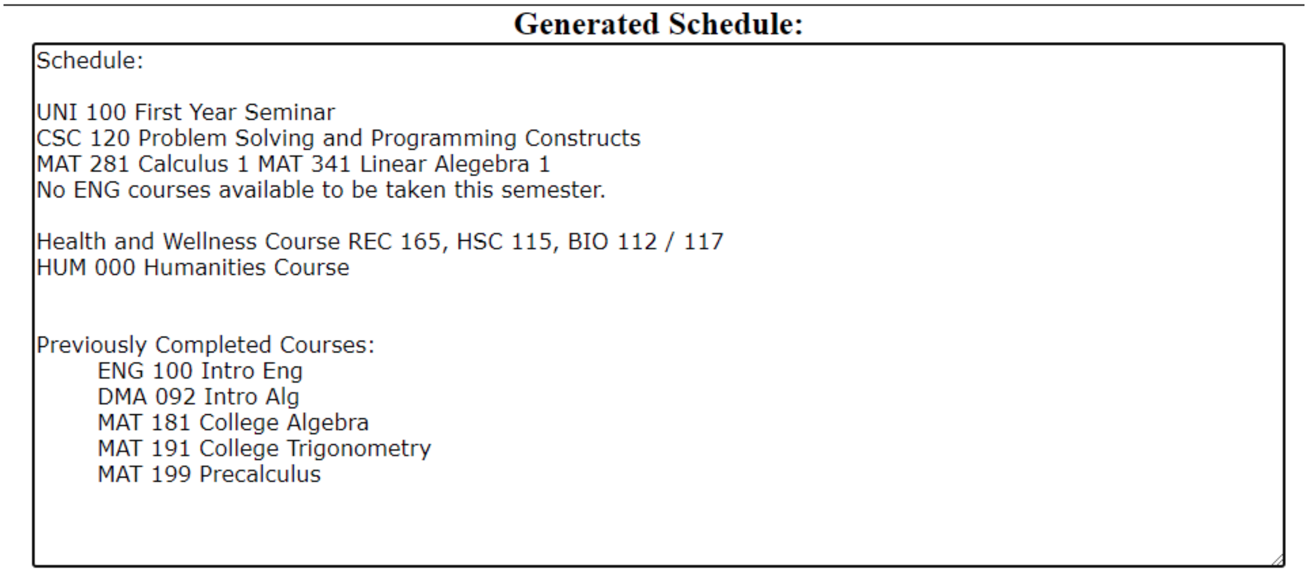


Additional Electives: (Counted towards degree)
☑ College Alg (MAT 181)
☑ College Trig (MAT 191)
☑ Precalculus (MAT 199)

☑ Intro Alg (DMA 092)
☑ English Language Skills (ENG 100)

If the user selects the course that should be taken before the suggested course, the rest will be checked off automatically. These checkboxes can be seen in the image below.

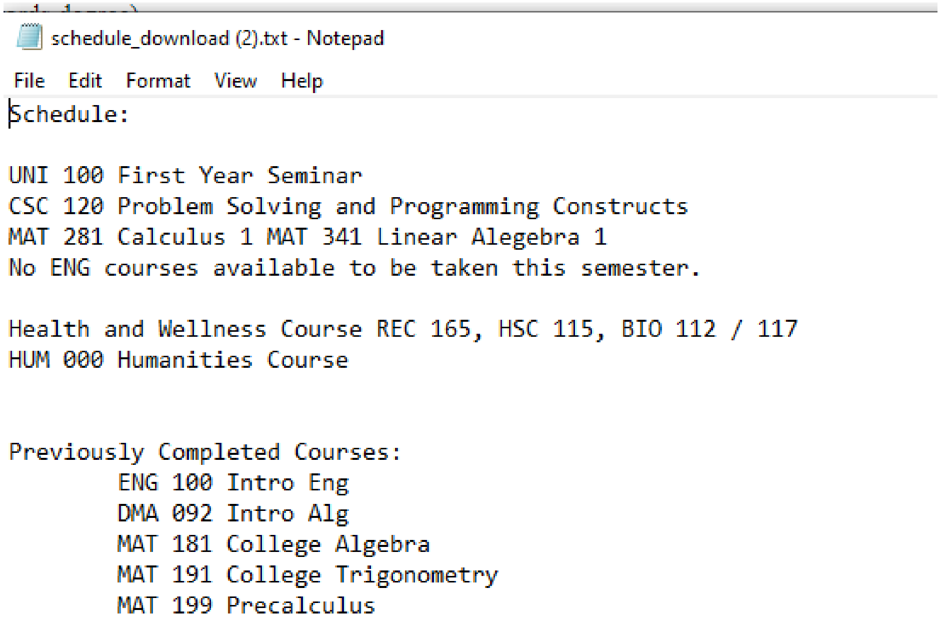
From here the user is ready to generate their schedule by scrolling to the bottom of the page and clicking on the “Update Schedule” button. The result of this can be seen below. As you can see, the text area will contain the scheduled classes, a warning for any classes that could not be scheduled this semester, and a list of all previous courses taken, or that credit has been given for because the user did not need to take them based on SAT/ACT and placement exam scores. The user can drag the bottom right corner to resize the schedule text area to any size they want.



Generated Schedule:

```
Schedule:

UNI 100 First Year Seminar
CSC 120 Problem Solving and Programming Constructs
MAT 281 Calculus 1 MAT 341 Linear Alegebra 1
No ENG courses available to be taken this semester.

Health and Wellness Course REC 165, HSC 115, BIO 112 / 117
HUM 000 Humanities Course


Previously Completed Courses:
        ENG 100 Intro Eng
        DMA 092 Intro Alg
        MAT 181 College Algebra
        MAT 191 College Trigonometry
        MAT 199 Precalculus
```

Additionally, the user can click the “Save Schedule” button to download the contents of the text area as a .txt file. This is what the user should keep and save somewhere to track what they are scheduling and what their progress is. However, if the user does not have it, it is no issue as all of the data can be reentered by checking off all of the buttons corresponding to the courses they have taken. As you can see below, the contents can be opened in any text editor. The file will always be downloaded with the name “schedule_download.txt”.


schedule_download (2).txt - Notepad
File Edit Format View Help

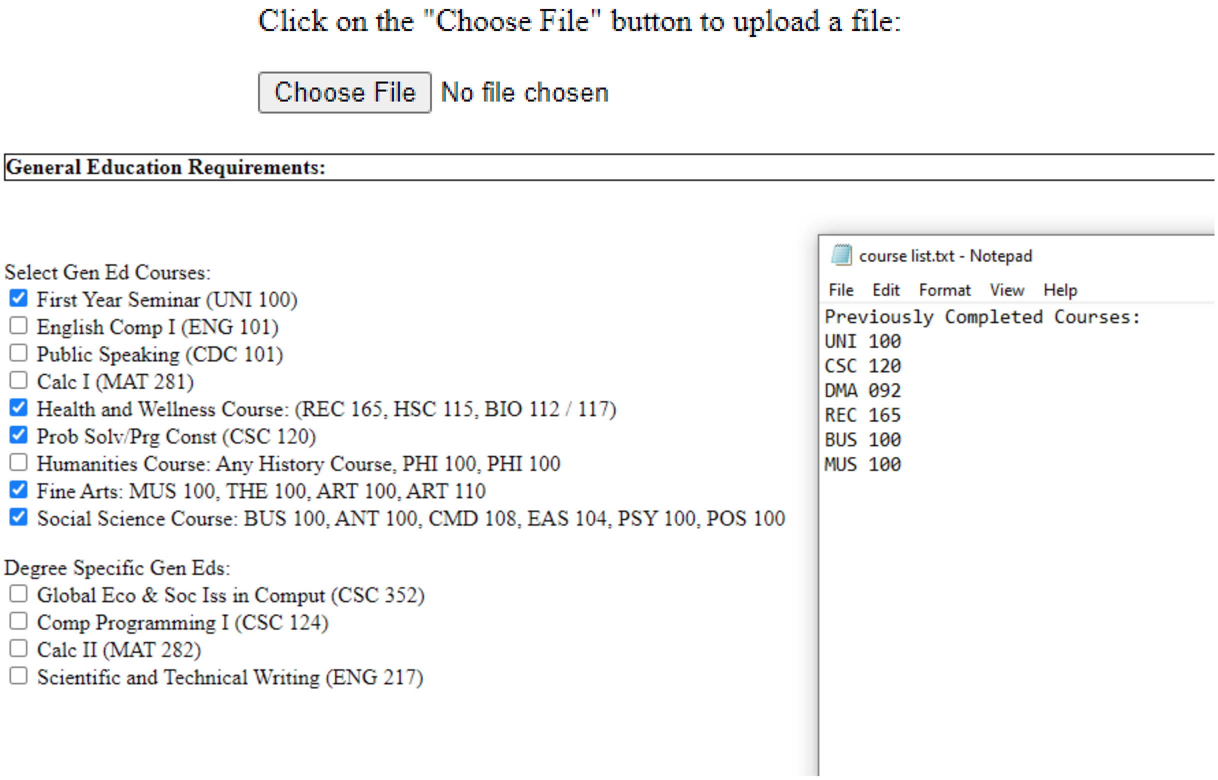
```
Schedule:

UNI 100 First Year Seminar
CSC 120 Problem Solving and Programming Constructs
MAT 281 Calculus 1 MAT 341 Linear Alegebra 1
No ENG courses available to be taken this semester.

Health and Wellness Course REC 165, HSC 115, BIO 112 / 117
HUM 000 Humanities Course


Previously Completed Courses:
        ENG 100 Intro Eng
        DMA 092 Intro Alg
        MAT 181 College Algebra
        MAT 191 College Trigonometry
        MAT 199 Precalculus
```

This can also function very well as the file the user will upload when they go to schedule for the next semester. We recommend that the user copies the scheduled courses into the "Previously Completed Courses" section for readability and to maintain continuity throughout their files. This is not required though, so if the student forgets there will not be any errors that come from this. After this point, the new student is now a returning user. Next we will discuss their usage cases and functions.

User: Returning User

______A returning user will typically be a user who has taken courses offered at Cal U before and is likely in the computer science major. Many of the use cases this user will have are similar to the cases of a faculty member or academic advisor. For this reason we will describe what they will typically be doing and provide instructions for it in the following section.

If the user still has access to the file they created before, then all they will have to do is upload the file to the site. From there they will verify that all classes they have taken were in the file they uploaded. The user is now ready to select the semester that they are scheduling for by selecting the correct checkbox, and then clicking on the “Update Schedule” button that was shown and discussed previously. As you can see below, there is the “Choose File” button. By clicking this the user will have their file explorer opened to select the .txt file to input. It will only accept .txt files and display an error if any other file types are entered. Also seen below is the open .txt file with the list of previously completed courses, and to the left there is the checkboxes that have been checked off based on the contents of the file (DMA 092 is not included because it is at the bottom of the screen so it was cropped off, but it is checked off). Now all the user will have to do is click the “Update Schedule” button and they will have their schedule generated for them.



Click on the "Choose File" button to upload a file:

No file chosen

General Education Requirements:

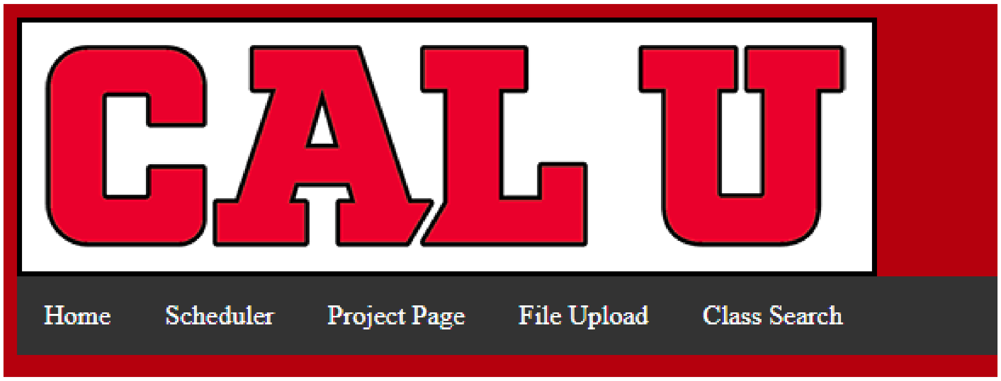
Select Gen Ed Courses:
☑ First Year Seminar (UNI 100)
☐ English Comp I (ENG 101)
☐ Public Speaking (CDC 101)
☐ Calc I (MAT 281)
☑ Health and Wellness Course: (REC 165, HSC 115, BIO 112 / 117)
☑ Prob Solv/Prg Const (CSC 120)
☐ Humanities Course: Any History Course, PHI 100, PHI 100
☑ Fine Arts: MUS 100, THE 100, ART 100, ART 110
☑ Social Science Course: BUS 100, ANT 100, CMD 108, EAS 104, PSY 100, POS 100

Degree Specific Gen Eds:
☐ Global Eco & Soc Iss in Comput (CSC 352)
☐ Comp Programming I (CSC 124)
☐ Calc II (MAT 282)
☐ Scientific and Technical Writing (ENG 217)

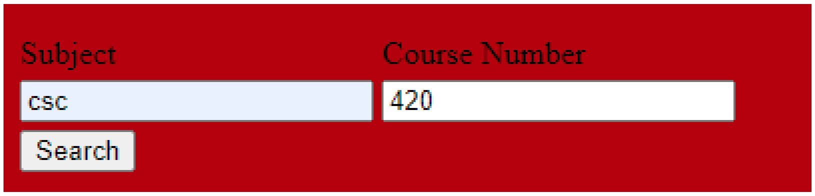
```
Previously Completed Courses:
UNI 100
CSC 120
DMA 092
REC 165
BUS 100
MUS 100
```

Usages: Database Searching

One of the advantages of the creation of our MySQL database, is that we can easily implement a function for the user to search the database for courses, and information about the course such as meeting times, and the semesters the course is offered. This can be useful to a first semester freshman and a senior in their final semester. To get to the search page, the user must click on the “Class Search” button on the navigation bar that is present on all our sites pages. This bar can be seen below the Cal U image.

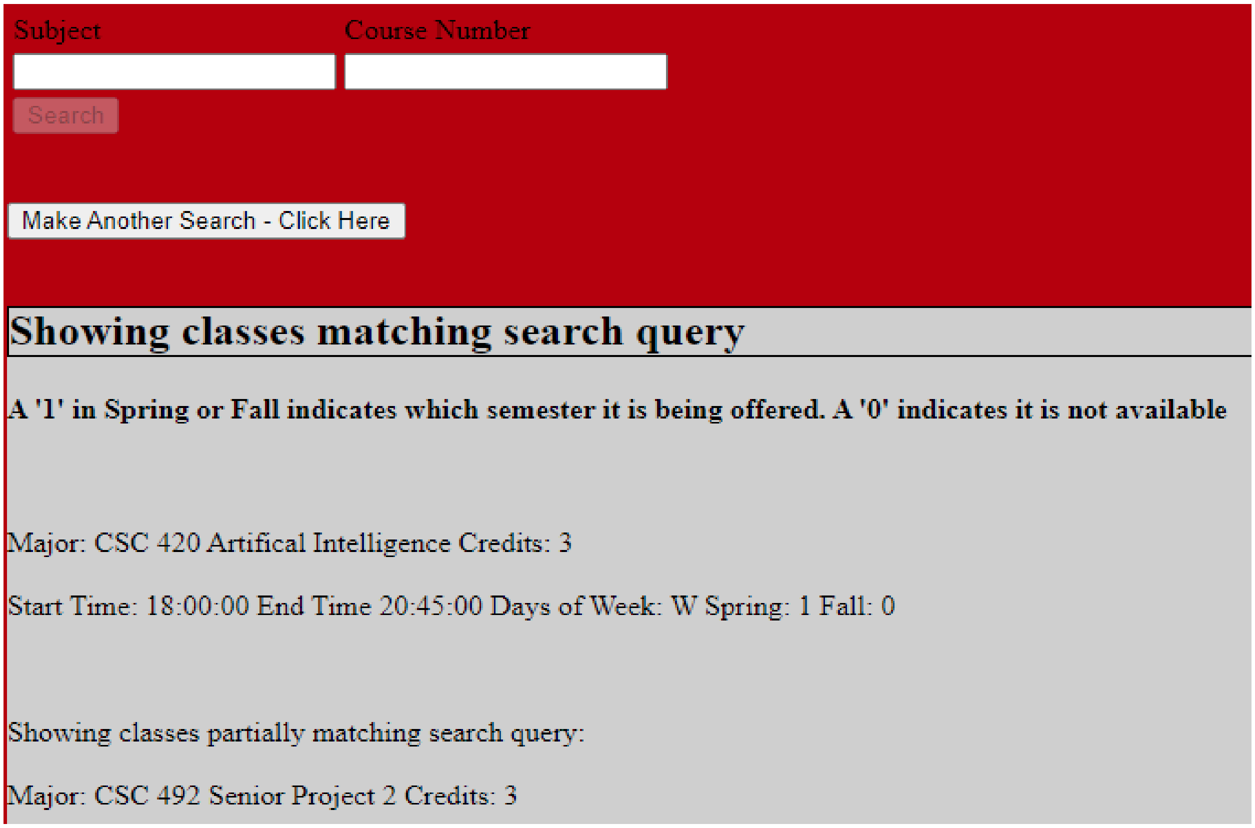


Once the user has clicked on it, they will be taken to a page where there are two text fields and a “Search” button. The user must enter a valid major offered at Cal U, and a course number that corresponds to the course they are trying to search for and obtain information on. An example of what a user searching for “CSC 420” will look like is pictures below, along with the results. It is important that you do not enter a space after or at the start of the data being put into the query (“csc” is valid, and “csc “ is not).



Subject: csc

Course Number: 420



Subject

Course Number

Showing classes matching search query

A '1' in Spring or Fall indicates which semester it is being offered. A '0' indicates it is not available

Major: CSC 420 Artifical Intelligence Credits: 3

Start Time: 18:00:00 End Time 20:45:00 Days of Week: W Spring: 1 Fall: 0

Showing classes partially matching search query:

Major: CSC 492 Senior Project 2 Credits: 3

Pictured above is the result of the search. From here the user can see the result of the search query. We also display a list of partial matches to what the user entered. These will be limited to courses in the same major that they entered in the "Major" field. They are not all displayed in the image above as it would be very large. The start and end times of the course is shown, the days of the week the class meets, and Spring/Fall is indicated with a 0 or 1, with 1 representing the semester it is offered. To make another search, the user must click the "Make Another Search" button. This will take them back to the "Class Search" page they were just on. The reason they must go to another page is so that they will not have any old query results stored before making another search.

<u>References:</u>

Bhatia, A. B., & Bansal, V. (2015). *Database management system*. Alpha Science International Ltd.

“HTML Element Reference.” *HTML Reference*, www.w3schools.com/tags/default.asp.

“MySQL Workbench.” *MySQL*, dev.mysql.com/doc/workbench/en/.

“MySQL Documentation.” *MySQL*, dev.mysql.com/doc/.

“PHP: Introduction.” *Php*, www.php.net/manual/en/intro.spl.php.

“Web Technology for Developers.” *MDN*, developer.mozilla.org/en-US/docs/Web/HTML/.

“Web Technology for Developers.” *MDN*, developer.mozilla.org/en-US/docs/Web/CSS.

“CSS Tutorial.” *W3Schools*, 1999, www.w3schools.com/css/default.asp.

“HTML Tutorial.” *W3Schools*, 1999, www.w3schools.com/html/default.asp.

Appendix

Appendix I: Technical Glossary

Administrator - A faculty member that updates and makes changes to the program

Cal U - California University of Pennsylvania.

SAT - Scholastic Assessment Test.

User - Students or faculty that use the program to generate schedules.

Javascript - A program that adds backend functionality to a website.

HTML - A coding language for basic website functionality and creation.

PHP - A program that adds frontend functionality to a website.

CSS - A coding language that goes hand to hand with HTML to create a website and give visual customization to the website.

Text File - A file containing only pure text, typically with a ".txt" file extension.

Algorithm - a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.

Degree Works - A web service offered at Cal U to assist the student in monitoring their grades and degree progress.

VIP - An online portal that offers many services to students at Cal U. These services include paying a bill, choosing housing, scheduling courses, and accessing many other essential services at Cal U.

Academic Advisor - A faculty member at Cal U who advises students what courses they should take next ino order to make the most progress towards their degree while fulfilling a student's specific needs.

Web Hosting - A service that hosts the files we have created to be accessible on the web by the general public.

Gear Host - A web hosting service that is free and allows us to upload the MySQL database files, and have the webpages communicate with them

Wix - An online web service that allows users to create simple web pages that can link to other web pages and services.

Database - a structured set of data held in a computer, especially one that is accessible in various ways.

Search Query - A web search query is a query based on a specific search term that a user enters into a web search engine to satisfy their information needs.

Appendix II: Team Details and Individual Contributions

______Our team did not have an easy way to meet with each other in person in a safe manner, so we all worked remotely throughout the past two semesters. Because of this we had some obstacles that came up. However, we were able to overcome them with clear communication and flexibility in our schedules. We consulted our Gantt chart at multiple points throughout the implementation phase to make sure we were on track and could see where we might be behind.

The team would present our weekly progress each Monday at 11:00 am. After that we would meet almost every other day throughout the week on Discord. Typically we would meet on Wednesday at 2:30 to discuss what changes needed to be made based on our feedback from the presentation, and what we are going to accomplish this week. Then we would meet on Friday to discuss our progress on what we discussed on Wednesday. On Sunday we would meet and create our weekly report and presentation for the following day, and discuss what we wanted to emphasize in that presentation.

Ryan Merow: Ryan lead the weekly progress presentations. He developed the initial webpages for the website and made several CSS styling iterations. He also assisted in accumulating the list of courses needed for database entry, as well as in testing the various parts of the database and website.

Londen Haskins: Londen worked on creating the database and initial functionality of the website. He set up web hosting for the site and server space for the database. He worked on site structure with the CSS file and navigation

Joseph Andrews: Joseph worked on getting most of the classes into the database. He created the file uploader for the schedule. He got the file reader to work and created the method that read the data from the text file and had it mark off the correct checkboxes.

Taylor Petricca: Taylor worked on the design of the scheduler in the fall semester, providing initial mockups and sketches for the scheduler functionality. In the spring semester he worked to add classes, class information, and information on majors into the database. Taylor also helped to create SQL queries to allow for accessing specific parts of the database.

Justin Proco: Justin worked on making the scheduler webpage and most of its functions. He helped make the file uploader for the .txt file. The layout of the webpage, including the buttons and information on the pages. The scheduler functions to generate the schedule and making it so all of the buttons had data that was retrieved from a query into the database. The download button to download the contents of the text area as a .txt file. And worked on an automatic checker function that will check off the courses prerequisite courses when the button is pressed.

Appendix III: Workflow Authentication

I, _Joseph Andrews__**(printed first name & last name)** hereby attest that I_Joseph Andrews__ **(signature)** have worked to both create and review this document, and approve of all the information and specifics detailed within. Signed on _04/25/2021_ **(date signed)**

I, _Taylor Petricca___**(printed first name & last name)** hereby attest that I_Taylor Petricca_ **(signature)** have worked to both create and review this document, and approve of all the information and specifics detailed within. Signed on __04/25/2021___ **(date signed)**

I, __Ryan_Merow___**(printed first name & last name)** hereby attest that I _Ryan_Merow_ **(signature)** have worked to both create and review this document, and approve of all the information and specifics detailed within. Signed on __4-25-21__**(date signed)**

I, _Londen Haskins__**(printed first name & last name)** hereby attest that I__*Londen Haskins*__ **(signature)** have worked to both create and review this document, and approve of all the information and specifics detailed within. Signed on __4/25/21__ **(date signed)**

I, _Justin Proco__**(printed first name & last name)** hereby attest that I_Justin Proco_ **(signature)** have worked to both create and review this document, and approve of all the information and specifics detailed within. Signed on __04/25/2021___ **(date signed)**

Appendix III: Code Listing