**AutoAdvisor User Guide for Windows**

**Installation/Set-Up**

Before we get started, ensure that you have the following:

* Microsoft Edge
* Edge Webdriver: [Microsoft Edge WebDriver - Microsoft Edge Developer](https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/)

After downloading the Edge Webdriver, you will need to modify your system environment variables to add the folder containing the Webdriver to PATH. More details can be found here: [How to Edit Your System PATH for Easy Command Line Access in Windows (howtogeek.com)](https://www.howtogeek.com/118594/how-to-edit-your-system-path-for-easy-command-line-access/)

After that. Grab everything from this github repository: [tblu4982/autoadvisor: Grabs student's academic transcript from VSU Banner and generates an advisory report (github.com)](https://github.com/tblu4982/autoadvisor). Ensure that the python files are all stored in the same file directory.

A screenshot of a computer

Description automatically generated

If you have saved the repository as a ZIP file, then you will need to unzip it. Open up a file explorer, go to your Downloads folder (or whichever folder you chose to save the zipped folder to), right click the file and select the ‘Extract’ option.

\*\*NOTE FOR MAC USERS:\*\* Some modifications to the code will be necessary to get it to work (otherwise, subsequent uses after the 1st will result in a segmentation fault). Open ‘project.py’ on your chosen Python IDE and do the following:

1. Within the ‘credentials’ class, find the ‘get\_vnums” function (Line 146).
2. Erase line 148
3. Erase lines 150 – 165
4. Remove two indentations from lines 150 – 156
5. Remove an indentation from lines 148 & 149
6. On line 149, replace function call ‘askopenfilename’ with a hardcoded path to your vnums text file

Open a text editor and create a text file (file extension should end in ‘.txt’). Fill that text file with a list of V-Numbers for which you’d like to generate transcripts for. It should look like this:

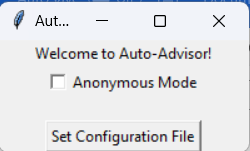
Text

Description automatically generated

NOTE: There is currently a minor bug in which AutoAdvisor fails to grab the last V-Number in the text file. A workaround is to simply add a dummy string to the bottom of the list.

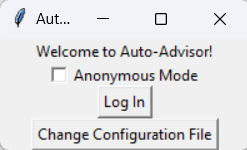
**Using the Program**

Now launch the program by running ‘project.py’. You should be greeted with this window:

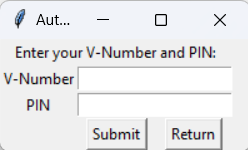


Anonymous Mode is for presentation purposes. It hides the name of the student from the transcript on the **Student** transcript (more will be explained on this later).

Once launched, AutoAdvisor requires a configuration file before you can proceed. Select the button and choose one of the configuration files that you have downloaded earlier (should be one of two excel files with TRANSCRIPT in file name). After you select a file, you should be greeted with the same window as before, but now an option to log in appears:



Select login, you should now be prompted to login. Enter the same login credentials that you would use to log in to VSU Banner:



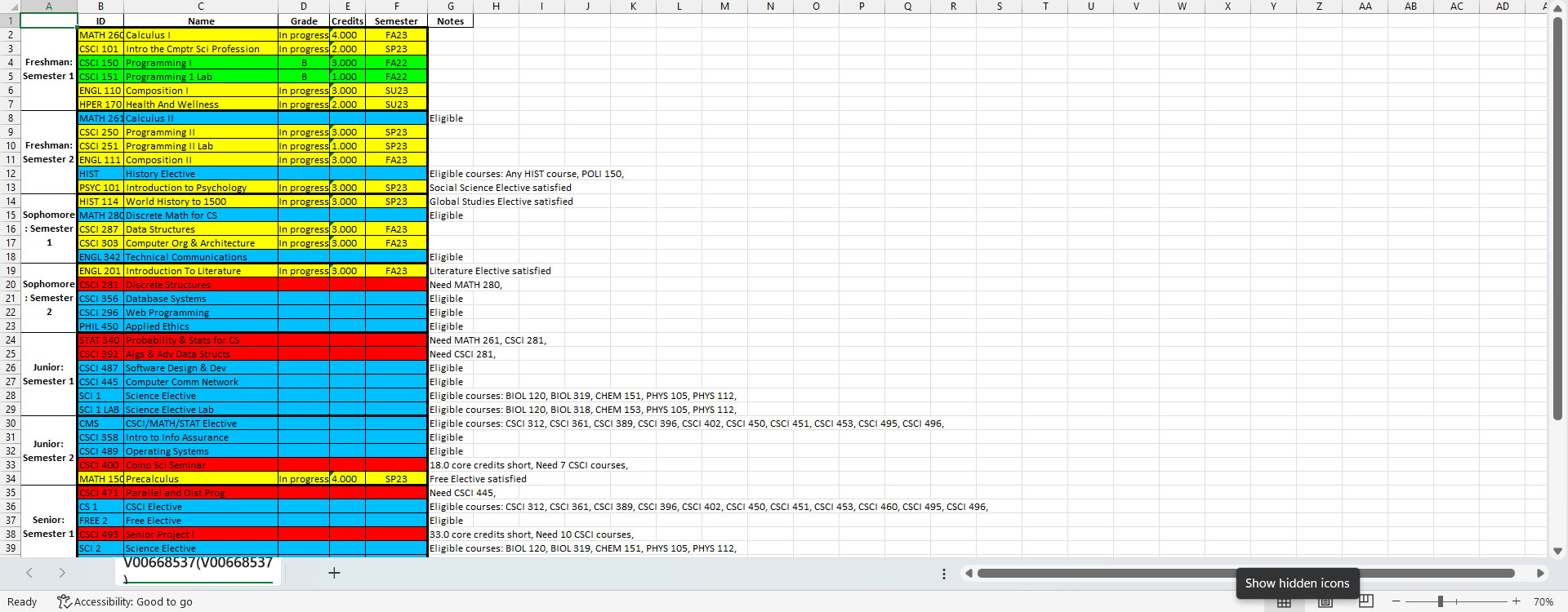
After you type in your credentials, click ‘Submit’. The window will close, and a terminal window will open, followed by an automated instance of Microsoft Edge. The webdriver will login using the login credentials that you provided, then it will prompt you to select a text file. Select the V-Numbers text file that you created earlier.

After the v-numbers file has been provided. The webdriver will crawl through banner and grab the transcripts of each student listed within the text file by their V-Number. If an invalid V-Number is detected, the program will alert you to which v-numbers it failed to grab at the end of execution.

AutoAdvisor will create a folder called ‘students’ in the same file directory that the executable is in, and for each student, it will create an additional folder within that ‘students’ folder with the student’s name (or V-Number is Anonymous Mode is toggled).

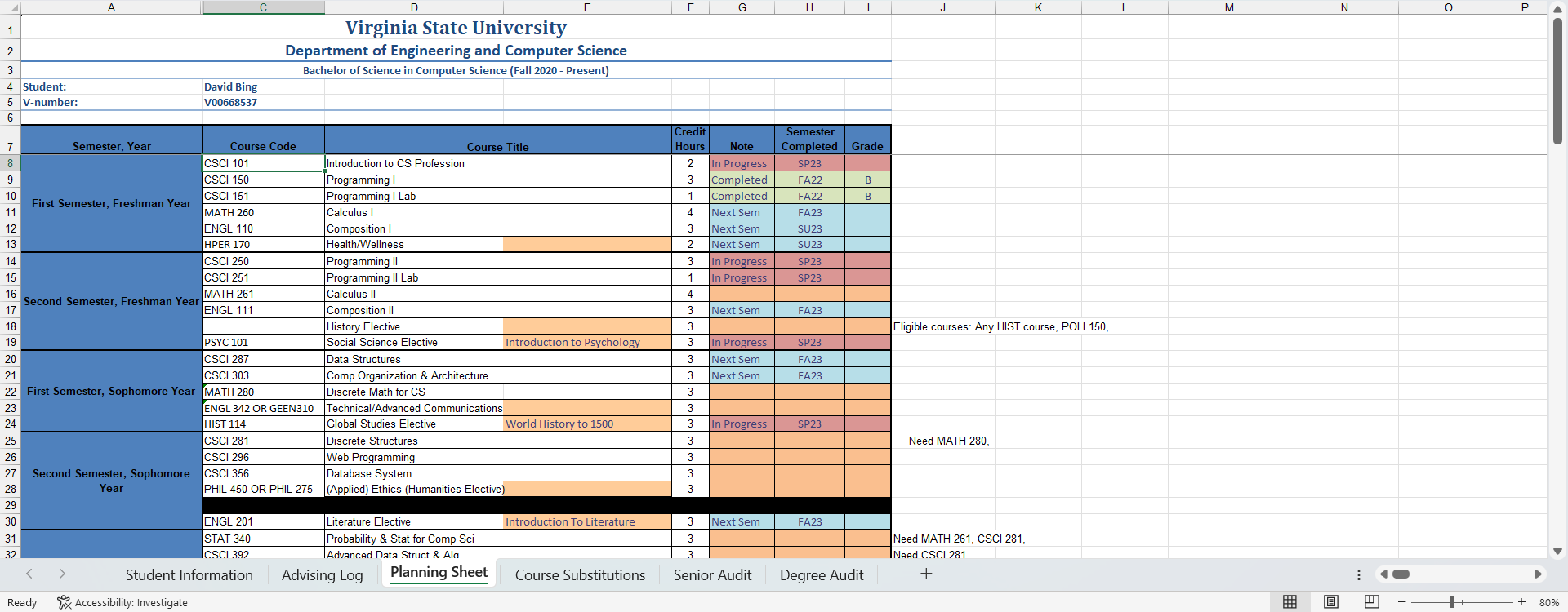
Inside each student folder are two text files containing raw data for all the courses that the student has taken (‘courses.txt’) and all the semesters where the student was enrolled (‘semesters.txt’). There are also two planning sheets, one labeled for **Advisors,** and one labeled for **Students**.

The **Students** planning sheet should be sent to the student. It contains a colorful planning sheet that contains comprehensive information which should assist the student in deciding what courses that they want to enroll for next semester.



* Courses labeled **Green** are completed courses.
* Courses labeled **Yellow** are courses that are either in progress, or a course that the student has enrolled for in a future semester
* Courses labeled **Blue** are courses that the student is eligible to enroll for. If the course is an elective, then it tells the student which courses can satisfy that elective.
* Courses labeled **Red** are courses that the student is ineligible to enroll for. It will also tell the student why they aren’t able to enroll for that course.

AutoAdvisor is configured to work seamlessly with the existing planning sheet provided for the Computer Science department (known as the **Advisors** planning sheet). The excel file ‘Template Panning Sheet’ contains the template that AutoAdvisor works with.



**The Configuration File**

Should the CSCI curriculum ever change, the configuration file can be modified to reflect those changes. Below is a configuration file that contains the 2020 CSCI curriculum:

Graphical user interface, application

Description automatically generated with medium confidence

Each Colum represents a different parameter:

* Column A represents the semester. Every course is grouped by the semester that a student is recommended to take that course in.
* Column B represents course type. AutoAdvisor generates a course structure for each course based on what type of course it is:
  + Core courses only have one course that can satisfy it, so AutoAdvisor will look for an exact match within the transcript
  + Elective courses have multiple courses that can satisfy it, so AutoAdvisor will look for any course that can satisfy that elective slot (more information provided in Column K)
  + Free electives are unrestricted, so AutoAdvisor will only populate it last after it has searched for core courses and electives
* Column C represents the course ID. In AutoAdvisor, each course ID represents a unique key within a Python dictionary (and thus **each course within configuration sheet must have a unique ID**). How AutoAdvisor handles this is based on what course type the course belongs to:
  + For core courses, AutoAdvisor will directly compare the key to course IDs within the transcript for an exact match
  + For free and elective courses, AutoAdvisor will not make this distinction, and instead search for a suitable course
* Column D represents the name of the course. This is a placeholder value that is used in the ‘Students’ version of the planning sheet where a course is missing.
* Column E represents the minimum passing grade. Courses are filtered out if they do not meet the minimum grade.
* Column F represents course credits. This is a placeholder value that is used in the ‘Students’ version of the planning sheet where a course is missing.
* Column G represents credits required. Courses that have a credits requirement will be labeled ‘ineligible’ if the student does not have the minimum credits required.
* Column H represents prerequisite courses. If the student has not taken a course, AutoAdvisor will first check to see if prerequisites have been satisfied. If there is a missing prerequisite, then the course will be labeled ‘ineligible’ with the missing course(s) being labelled in the Notes section.
  + When modifying this column contents, courses must be labelled by their course ID (ex: CSCI 101)
  + If a course has multiple prerequisites, then each course must be separated by a comma (ex: CSCI 150, CSCI 151)
  + If a course requires a certain number of specific courses to be taken (ex: Senior Project I or Computer Science Seminar), then you must label the prerequisite like this: CSCI\*11
    - The CSCI part tells AutoAdvisor to look for CSCI courses
    - The \*11 part tells AutoAdvisor that it is looking for 11 courses
* Column I represent corequisites. This column currently has no function. **DO NOT DELETE!**
* Column J represent course substitutes. For certain core courses, there are courses that are commonly used as a substitute. AutoAdvisor will first look for the core course, and if the course is missing, then it looks for a substitute course. If a substitute course is used, then it will be identified as such within the Notes section
* Column K represents eligible courses. For elective courses, a list of courses can satisfy them.
  + Each course must be separated by comma
  + If any course from a department can be used to satisfy an elective, then you can just list the four-letter character ID for that department instead of individually labeling each specific course (ex: SOCI, FREN, GERM)
* Column L represents eligible course prerequisites. Some courses that can be used to satisfy an elective may have prerequisites. If they do, then they must be listed here:
  + Each entry must be enclosed in a set of tags ‘<>’
  + Inside those tags, list the course and its prerequisites (ex: <COURSE: PREQUISITE>)
  + Each entry must be separated by commas (ex: <C1: P1>, <C2: P2>)