

Navigating the curriculum at William & Mary can be a complex and often overwhelming experience for students, particularly those trying to map out their entire four-year path. Between general education requirements, major and minor requirements, credit minimums, and options like study abroad, it's easy to lose track of what is required and when to take it. For my final project, I developed a four-year academic planning tool that helps incoming William & Mary students generate customized semester-by-semester course schedules (an image of a schedule can be seen in Figure I of the Appendix). The goal of this tool is to simplify the academic planning process by producing a realistic, color-coded schedule that reflects students' intended majors, minors, preferences, and incoming credits.

The application is built using Dash and allows users to input a primary major, an optional secondary major or minor, the number of credits they enter with, whether they plan to study abroad, and what language they plan to study. Based on this information, the app generates a visual plan that distributes the necessary coursework over eight semesters, taking into account prerequisites, credit limits, and requirements for general education and graduation. The plan includes both required courses and electives, and it adapts to different types of students by offering flexible combinations of programs. For example, a student might choose to major in Government, minor in History, and study abroad. The resulting schedule accounts for all of these preferences while still meeting COLL requirements and total credit hours.

To build the app, I relied on two key sources of data. First, I scraped course information from the official William & Mary catalog<sup>1</sup> using Selenium and BeautifulSoup4<sup>2</sup>. This gave me a structured dataset that includes course codes, credit hours, COLL designations, and prerequisite information. This course catalog data is essential for ensuring the app doesn't assign a course before its prerequisites are completed, and that it recognizes which courses satisfy COLL requirements. Figure 2 in the appendix contains a sample of this dataset. Second, I compiled

---

<sup>1</sup> <https://catalog.wm.edu/>

<sup>2</sup> These are very common web-scraping packages in python. They go onto the website hosting the data and are able to click on links and extract data from specific parts of the HTML code that correspond to specific pieces of information such as course code, credits or prereqs

major and minor requirements manually for the ten most popular major and minor programs. This was necessary because the requirements listed on department websites are written in inconsistent formats that are hard to parse automatically.<sup>3</sup> I structured these requirements in nested dictionaries that indicate whether a student needs to take all of the listed courses or a selection from a longer list of options. A sample of this data structure is included in Figure 3 in the appendix. While this manual approach limits the number of majors currently supported, it allows for precise control over the logic used in the scheduling process, which is essential to the functionality of this project.

The core logic for building a student's schedule is handled in the `make_a_schedule` class. This class includes functions that identify which classes are needed for the student's chosen major(s) and minor, ensure that COLL Curriculum requirements are covered—including math proficiency, arts proficiency, foreign language proficiency, and domain-specific breadth—and check for any unmet prerequisites. It also tracks elective credits to help ensure the student reaches the 120-credit threshold for graduation and flags any remaining requirements that may need to be fulfilled to stay on track for degree completion.

Once the complete list of courses needed for a student's major(s), minor, and general education requirements is assembled, the program uses a combination of hierarchical<sup>4</sup> and topological<sup>5</sup> sorting to organize them in a logical and dependency-aware sequence. Courses are first categorized into four tiers: (1) prerequisite-only courses that serve as foundational building blocks; (2) courses that are both prerequisites and have their own prerequisites, which often sit at the heart of a major's structure; (3) advanced courses that depend on earlier prerequisites but do not serve as prerequisites themselves; and (4) electives or standalone COLL courses with no dependency constraints. To prevent logical errors, a topological sorting algorithm is applied within each tier to ensure that no course is scheduled before all of its prerequisites have been completed. This sorting step is essential before courses are assigned to specific semesters, as it respects the academic dependencies and guarantees that students progress through their degree in a coherent, requirement-aware order.

---

<sup>3</sup> Some majors have lists of required courses, others simply have a requirement to take  $n$  credits in the major while others have tracks. In addition many majors have distinct sections where courses are required and others where the student is to pick  $n$  courses from the following list.

<sup>4</sup> Organizing items into levels based on dependency or importance.

<sup>5</sup> Arranging items in a sequence where each item appears only after any items it depends on

Next, the program begins the scheduling process by iterating over each course and semester, ensuring that courses are placed correctly according to their prerequisites and credit limits. First, it places any course that must be taken in specific semesters, such as language courses, COLL 100/150, or study abroad. Once these initial courses are placed, the program iterates through each course from the sorted list and tries to place it. For each course, the program verifies that its prerequisites have already been scheduled in earlier semesters, respecting the proper learning sequence. It then ensures there is enough space in the chosen semester by trying to keep the credit load at 15 credits or fewer. If that's not possible, the program allows up to 18 credits per semester. By systematically evaluating each course's prerequisites, semester placement, and credit limits, the app creates a well-structured schedule that fits within these constraints.

Once the scheduling process is complete, the app generates a detailed, color-coded chart using matplotlib to represent each semester's course load visually. This chart is designed to act as a guide for students, helping them easily see which courses they should plan on taking each semester and how those courses align with their academic goals. The different colors in the chart help to distinguish between various types of courses, such as those required for the student's major, minor, COLL curriculum, and electives.

The user interface was designed with simplicity and usability in mind, allowing incoming students to explore different academic paths easily. It prompts students to input their preferences, such as their chosen major(s), minor, and any other academic preferences, and then dynamically generates a customized course plan based on those selections. The application allows students to experiment with different combinations of courses and academic tracks, giving them the flexibility to try out various scenarios and see how their schedule could look with various choices. For example, if a student selects a double major, the interface automatically hides the "minor" option to prevent confusion. This interactive design encourages students to explore a wide range of possibilities and gain a deeper understanding of their academic options. The interface is nice but simple; it leverages a Dash Bootstrap Components (DBC) theme.

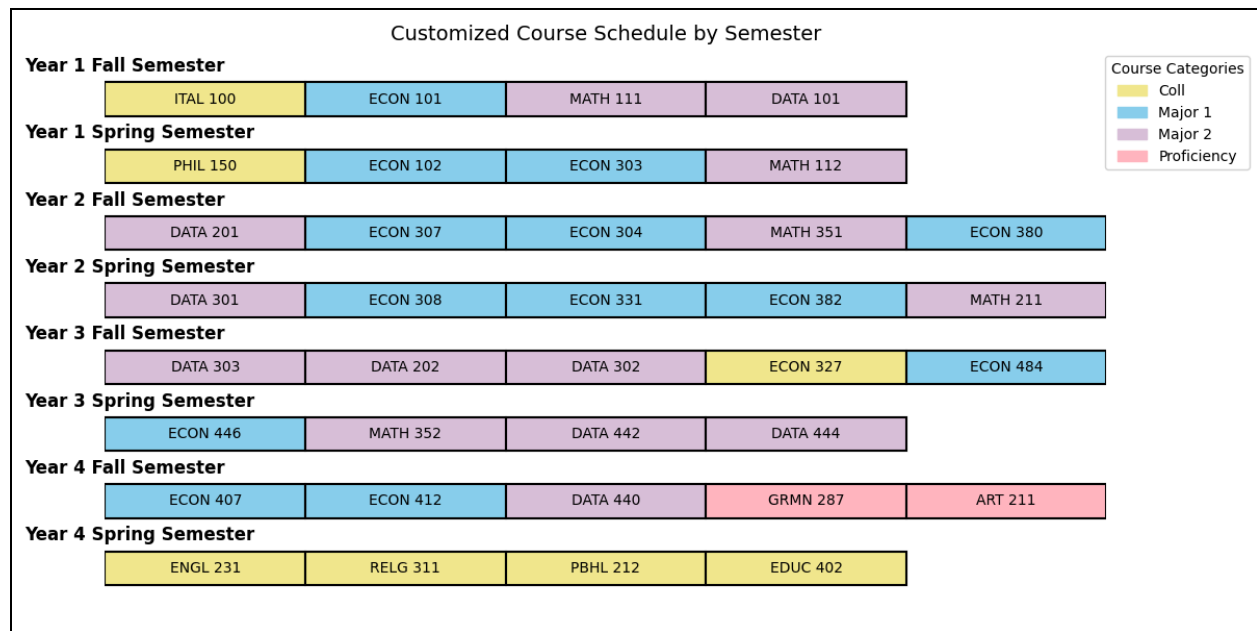
One of the most exciting parts of this project was the opportunity to translate abstract academic rules into a clear and concrete tool that students can use. However, the project does

have limitations. Because the major and minor requirements had to be entered by hand, the app currently only supports ten of the most popular majors and minors. Additionally, it does not yet account for semester-specific course availability or special cases like honors theses, COLL 400 seminars, or dual-crediting of classes for both majors and COLL. In future versions, I would like to expand the scope of majors and minors, allow students to mark certain courses as “must-take” or “preferred,” and integrate live course availability from Banner. I also hope to add a save feature that would let students return to their plans and adjust them over time.

Ultimately, this project taught me how to build a functional planning tool using real university data and Python. It required not only technical skills like web scraping, data structuring, and visualization, but also an understanding of the student experience and the institutional requirements that shape it. I hope that this tool, even in its current form, can make the planning process less stressful and more transparent for students at William & Mary and that it can be expanded upon to become more dynamic.

## APPENDIX:

**FIGURE I: Image of Schedule Generated**



**FIGURE II: Subset of the course catalog containing Government courses**

```
[{'course_code': 'GOVT 100', 'credits': 4, 'prereqs': None, 'coreqs': '', 'coll': 'COLL 100', 'domain': None},
{'course_code': 'GOVT 150', 'credits': 4, 'prereqs': None, 'coreqs': '', 'coll': 'COLL 150', 'domain': None},
{'course_code': 'GOVT 201', 'credits': 3, 'prereqs': None, 'coreqs': '', 'coll': None, 'domain': 'CSI'},
{'course_code': 'GOVT 202', 'credits': 3, 'prereqs': None, 'coreqs': '', 'coll': None, 'domain': None},
{'course_code': 'GOVT 203', 'credits': 3, 'prereqs': None, 'coreqs': '', 'coll': None, 'domain': 'CSI'},
{'course_code': 'GOVT 204', 'credits': 3, 'prereqs': None, 'coreqs': '', 'coll': None, 'domain': 'CSI'},
{'course_code': 'GOVT 301', 'credits': 3, 'prereqs': None, 'coreqs': '', 'coll': None, 'domain': 'CSI'},
{'course_code': 'GOVT 302', 'credits': 3, 'prereqs': 'GOVT 301', 'coreqs': '', 'coll': None, 'domain': None},
{'course_code': 'GOVT 305', 'credits': 3, 'prereqs': None, 'coreqs': '', 'coll': 'COLL 200', 'domain': 'ALV'},
{'course_code': 'GOVT 306', 'credits': 3, 'prereqs': None, 'coreqs': '', 'coll': None, 'domain': None},
{'course_code': 'GOVT 307', 'credits': 3, 'prereqs': None, 'coreqs': '', 'coll': 'ACTV', 'domain': None},
{'course_code': 'GOVT 310', 'credits': 3, 'prereqs': None, 'coreqs': '', 'coll': 'COLL 200', 'domain': 'CSI'},
{'course_code': 'GOVT 311', 'credits': 3, 'prereqs': 'GOVT 203', 'coreqs': '', 'coll': None, 'domain': None},
{'course_code': 'GOVT 312', 'credits': 3, 'prereqs': 'GOVT 203', 'coreqs': '', 'coll': None, 'domain': None},
...]
```

**FIGURE III: Sample of Majors data structure**

```
"government": { "requirements": [  
  
  {"type": "all_of", "courses": ["GOVT 201", "GOVT 203", "GOVT 204", "GOVT 301"]},  
  
  {"type": "choose_n", "n": 1, "groups": [  
  
    {"type": "any_of", "courses": ["GOVT 202", "GOVT 305", "GOVT 392"]},  
  
    {"type": "any_of", "courses": ["GOVT 401", "GOVT 402", "GOVT 403", "GOVT 404",  
    "GOVT 433", "GOVT 435", "GOVT 440", "GOVT 454", "GOVT 455", "GOVT 465",  
    "GOVT 470", "GOVT 482", "GOVT 489", "GOVT 491", "GOVT 493", "GOVT 494",  
    "GOVT 495", "GOVT 496", "GOVT 498"]} ]},  
  
  {"type": "choose_n", "n": 5, "groups": [  
  
    {"type": "any_of", "courses": ["GOVT 100", "GOVT 150", "GOVT 302", "GOVT 305",  
    "GOVT 306", "GOVT 307", "GOVT 310", "GOVT 311", "GOVT 312", "GOVT 322", "GOVT  
    324", "GOVT 235", "GOVT 326", "GOVT 327", "GOVT 328", "GOVT 329", "GOVT 330",  
    "GOVT 334", "GOVT 335", "GOVT 336", "GOVT 337", "GOVT 338", "GOVT 339", "GOVT  
    340", "GOVT 350", "GOVT 351", "GOVT 352", "GOVT 353", "GOVT 355", "GOVT 360",  
    "GOVT 361", "GOVT 370", "GOVT 372", "GOVT 373", "GOVT 374", "GOVT 388", "GOVT  
    389", "GOVT 390", "GOVT 391", "GOVT 392", "GOVT 393", "GOVT 394", "GOVT 401",  
    "GOVT 402", "GOVT 403", "GOVT 404", "GOVT 433", "GOVT 435", "GOVT 440", "GOVT  
    454", "GOVT 455", "GOVT 465", "GOVT 470", "GOVT 482", "GOVT 489", "GOVT 491",  
    "GOVT 493", "GOVT 494", "GOVT 495", "GOVT 496", "GOVT 498"]} ]}  
  
  ]},  
}
```

**Figure IV: Dash App Webpage:**

## Custom William & Mary Four Year Plans

Welcome! Use this tool to create a personalized four-year plan. Start by entering your credits, selecting your primary major, and customizing additional preferences below.

### Input Preferences Here:

Provide your academic and study-abroad preferences. Depending on your degree type, additional fields will appear dynamically to guide your selections.

Select Primary Major:

Choose... ▾

Enter Number of Credits Already Earned:

Select Language:

Choose... ▾

NOTE: International Relations Majors must select a language even if they have already met the FLP requirement.

Semester Abroad Preference:

Choose... ▾

Choose Degree Type:

☒ Single Major

☐ Double Major

☐ Major and Minor

Submit