DegreeFlow

Group 4

GitHub repo: https://github.com/Posxidon/Degree-Flow.git

Brief Overview of Achievements:

Our project, DegreeFlow, is a web-based academic planning tool designed to assist students in navigating their degree requirements efficiently. Throughout the project, we successfully developed a secure login system, integrated seat alert functionalities, built a transcript parsing tool, and implemented a partial course search and scheduling feature. While we encountered some limitations due to external dependencies, we adapted our goals and focused on delivering core functionalities with the available resources.

P0 and P1 Requirements & Completion Status:

- Course Search: 30%
- Schedule Generation: 50%
- Prerequisite Verification: 0%
- Integration with Mosaic: 0%
- Transcript Parsing: 90%
- Secure Login: 90%
- Seat Alert: 100%
- Real-Time Notifications: 0%

Explanation for Requirements Below 80%:

Course search

This feature was very limited due to the constraints of the university's API. Filters such as course difficulty were not approved for use due to risk involved in opinion based rating systems.

Schedule Generation:

Full schedule generation requires access to real-time class schedules, which we were not provided. UTS instructed us that this information was restricted and unavailable to student developers.

Prerequisite Verification:

While we have basic prerequisite information displayed, we were unable to verify or enforce prerequisites automatically. The necessary data was inaccessible, and internal confusion within UTS made us unable to get help, resulting in this feature being unfeasible.

Mosaic Integration:

Access to Mosaic API was not possible as the university informed us the process was long and not guaranteed. Due to time constraints, we decided against it.

Real-Time Notifications:

Required data was not available due to university security constraints, making this feature unfeasible.

What Worked Well:

- Secure login and user flow
- Transcript parsing and tracking
- Seat alert system

What Did Not Work:

- Mosaic integration due to lack of API access
- Real-time notifications due to data access restrictions
- Full course search and prerequisite verification due api documentation difficulties

What We Learned:

Institutional Communication Can Be a Major Bottleneck:
Delays in responses and limited cooperation from university departments affected our ability to implement several features. Relying on multiple departments made development timelines unpredictable and led to dropped many features.

The Importance of Documentation:
The Undergraduate Calendar API lacked sufficient documentation, requiring us to invest extra time to reverse-engineer endpoints and understand responses. Furthermore, those regularly worked with the API were unable to help because they also lacked sufficient understanding of the API.

Continuous Integration is Essential:
Working in separate branches without frequent merges led to major conflicts and rework.

Regular Meetings Improve Productivity:
Frequent check-ins kept everyone informed, aligned, and focused, especially as the complexity of the project increased.

Planning Beyond Scope Introduces Risk:
We underestimated the difficulty of having multiple systems and components communicate with each other. A failure in any one part of the communication chain could break the system, and coordinating these parts proved harder than expected.

Client Needs Must Stay Front and Center:
As we moved deeper into development, it became easier to lose sight of the client's priorities. Institutional restrictions (security, data access) further limited what we could do, forcing us to realign our goals with what was realistically achievable.