Plan: FIT Schedule Planner

by: Pedro Moura, Jordan Synodis

Project Goal and Motivation

- **Provide students with a user-friendly interface for streamlined class registration.**
- **Combine access to class schedules, degree evaluations, and program requirements in one place.**
- **Minimize confusion and misunderstandings in the class registration process.**
- **Help students avoid unnecessary classes and scheduling difficulties.**
- **Enhance efficiency by offering a comprehensive tool for academic planning.**

Approach

- Develop a user-friendly interface that streamlines the class registration process for students at FIT.
- Create a tool that aims to combine access to class schedules, degree evaluations, and program requirements in one convenient location.
- Implement features such as class search filters, schedule visualization, and progress checklists are key components of the development approach.
- The user will be able to add classes to their schedule, check prerequisites, identify overlapping classes, and utilize filters for a tailored search experience.
- Integrate personal time blocks to prevent class schedule conflicts with extracurricular activities.

Novel Features/Functionalities

01 **Streamlined Web Interface**

- Combines class schedules, degree evaluations, and program requirements
- Enables seamless class registration

02 **Class Search Filters**

- Find classes by subject, professor, times, course number, credit range, or RateMyProfessor score

03 **Schedule Visualization**

- View weekly schedules as classes are added or removed

04 **Progress Checklist**

- Generated from CAPP Degree Evaluation
- Tracks program completion progress

05 **User-Friendly Interface**

- Consistent layout, navigation bar, and FAQs tab
- Pop-ups and loading bars for real-time feedback and status updates

Algorithms and Tools

- **Python**
  - Programming language
- **Scapy (Python)**
  - Allow us to scrape information from the internet
- **Django/Flask**
  - Framework to build our web application
- **Render.com**
  - Web Application deployment
- **https://apps.fit.edu/schedule/main-campus/fall**
  - Data source for webscraping fall classes
- **https://apps.fit.edu/schedule/main-campus/spring**
  - Data Source for webscraping spring classes
- **https://www.fit.edu/programs/**
  - Degree requirements data source

Technical Challenges

- **Privacy could be a challenge since students' CAPP Degree Evaluations are password-protected.**
  - In the prototype stage of this project, students may be required to manually download their CAPP evaluation and providing it to the system during testing of the application

- **Distribution of the final application could be a challenge since the FIT Schedule Planner is intended to be a GUI. Our group is only familiar with creating GUI in executable files, but executable files are not compatible with Apple devices.**
  - Distribution may require the development of a web application in place of a GUI. A web application might be more convenient for users compared to a GUI anyway.

Milestone 1

- Compare and select technical tools for generating an interface, scraping information off the internet, developing the schedule planner, and developing the checklist.
  - Provide small (“hello world”) demos to evaluate such tools.
- Resolve technical challenges with privacy, distribution, and website creation.
- Compare and select collaboration tools for software development, documents/presentations, communication, and a task calendar.
- Create Requirement Document
- Create Design Document
- Create Test Plan

Milestone 2

- Implement, test, and demo traversing the interface
- Implement, test, and demo loading the CAPP Degree Evaluation
  - Include at least one test where the CAPP Degree Evaluation is reuploaded
- Implement, test, and demo accessing the program checklist
  - Include tests for a variety of different programs

Milestone 3

- Implement, test, and demo searching for classes
- Implement, test, and demo class filtering system
  - Include at least one test for each filter

- Implement, test, and demo adding a class to the schedule
  - Include at least one test where prerequisites are not met
  - Include at least one test where a time conflict is present
  - Include at least one test where the class is full

- Implement, test, and demo removing a class from the schedule