StudyBuddy: ECE Senior Design Proposal

The Problem

College students often waste time and money taking extra credits for their degree due to poor planning, lack of information and lack of sufficient guidance from academic advisors. A study by the Greater Texas Foundation found that extra credits costs went up to $120 million annually for students and taxpayers in Texas.[1] This indicates that the current method via which college students form a degree plan needs to be improved.

The Solution

StudyBuddy is a website application that aims to solve this problem by helping students come up with an optimal study plan geared towards their preferences and by providing a platform for them to seek advice from their academic advisor and upperclassmen.

Value Proposition	**App Features**
Ease of Planning and Communication	● Interactive UI that allows dragging & dropping courses from one semester to another ● Ability for advisors to edit study plans with their advice and display the changes made to the student ● Allows students to view good and bad study plans of alumni so they can learn from their mistakes
Optimal Plan	● Takes in student preferences such as: max number of credits for a semester, number of semesters to graduate, whether to sync study plan with a friend, etc.) and suggests a few optimal study plans that fit the criteria. Students can then make changes as they see appropriate. ● Allows easy selection of electives and gen eds

Other Features (if time permits):

- Notifications when spots open up for a class/section in your current study plan
- Integrate upperclassmen / alumni reviews for specific courses / professors to aid students in choosing which classes to take

---

1 https://hechingerreport.org/wp-content/uploads/2017/08/CCRC_GTF-StrengtheningTX-Transfer_Final-Report_Full-Report_Jan-2017.pdf

Implementation

- **Frontend:** We will most likely be using React for the frontend, as it is one of the most popular and easy to use framework with a lot of extendability with tools like TailwindCSS or MatrialUI to make the application visually pleasing and intuitive to use.

- **Backend:** Since our frontend framework is already in JavaScript (React), we are leaning towards using NodeJS backend frameworks such as ExpressJS. However, depending on how we choose to solve the problem we might also use Python backend frameworks such as Flask or FastAPI.

- **Database:** We will be using a NO-SQL database such as MongoDB, Firebase, or Cassandra depending on which service provides the most convenience for our requirements.

- **Web-hosting:** For web-hosting, we could use services such as Linode, Heroku, Vercel, or Github-pages

- **Cloud Services:** For future scalability we might choose to use cloud services such as GCP, AWS, or Cloudflare.

- **Authentication:** We aim to implement our own authentication with JWT, if this seems too hard to be implemented we will switch to Google OAuth. However, we prefer not to use Google OAuth as it authenticates using gmail addresses, but Purdue emails uses outlook

- **3rd party API:** We will check if there is a myPurdue API that can deliver us information regarding all classes. This would be preferable over web scraping. We found an unofficial API that we may be able to use but we need to look into it further.

Competitive Analysis

1. Purdue Course Catalog & MyPurduePlan

   Issues:
   - Requirements are scattered around and are hard to understand
   - Classes that count for special requirements are not accounted for in MyPurduePlan and must be confirmed with advisor
   - Study Plan is not tailored to the student's preferences
   - Experimental course options tend to be hard to find and are not advertised on the main page

- No easy way for back and forth communication between advisors and students when editing a study plan since UI is not interactive
- Bad UI makes it tedious for students to edit and visualize different plans
- Does not incorporate feedback from alumni / upperclassmen on study plans or specific courses
- Hard to figure out which electives / gen ed courses satisfy which requirement, students end up having to refer to multiple different web pages for this

Links to Documentation

- **Frontend**
  - React
  - TailwindCSS
  - MaterialUI

- **Backend**
  - ExpressJS
  - Flask
  - FastAPI

- **Database**
  - MongoDB
  - Firebase
  - Cassandra

- **Web-hosting**
  - Linode
  - Heroku
  - Vercel
  - GitHub-pages

- **Cloud Services**
  - GCP
  - AWS
  - Cloudfare

- **Authentication**
  - GoogleOAuth
  - JWT