

This project was bootstrapped with [Create React App](#).

Overview

This project, Schedule Builder, is a web application that provides an interface students to plan their courses for the upcoming semester.

File Structure

/api: Backend folder

 /calendar: Folder for calendar view

 /venv: Folder containing all venv files (Lib/site-packages, Scripts, homepage)

 /gitignore: Git ginore

 /app.py: Retrieves courses data retrieval from Flask

 /calendar_blueprint.py: Calendar blueprint data retrieval from Flask

 /database.py: File onnects to MongoDB server

 /requirements.txt: Docuemnt of backend requirements

 /web.py: Formats backend text retrievd from backend for courses

/frontend: Frontend folder

 /src: Source folder

 /app: App folder

 /build-calendar: Calendar folder

 /page.tsx: BuildCalendar function that creaes format of the front page

 /recommendation: Recomenndation folder

 /page.css: CSS for the recommendation page

 /pagetsx: Typescript for the recommendation page

 /globals.css: CSS for global

 /layout.tsx: Main page navbar

 /page.css: CSS for the navbar

 /page.module.css: CSS for main page

 /page.tsx: Typescript functions for the homepage

 /provider.tsx: Typescript file that imports Chakra

 /components: Components folder for diffferent Chakra elements

 /build-calendar: Folder of elements for calendar (build-calendar, header, homepage, layout, recommendation)

 /header: Folder of elements to navbar

 /homepage: Folder of elements for suggestions and watchlist

 /layout: Folder of elements for layout of the home page

 /recommendation: Folder of elements for recommendation (page, page.module.css, page.tsx, provider.tsx, globals.css, layout.tsx, header, build-calendar)

```
/recommendation: Folder of elements to elements for the recommendation page (comment, course, description, types)

/hook: Folder takes courses from the backend and populates them into course card

/mock: Folder that holds the parsed courses from the backend

/public: Folder with logo image

/services: empty folder

/test: Testing folder

/SearchResults.test.tsx: Tests for search results

/themes: Folder of of the day and night component

/types: Folder that holds the format of the scraped course data

/.eslintrc.json: Linting file

/.gitignore: Dependencies

/README.md: Read me file

/package-lock.json: Package JSON file

/package.json: Package JSON file

/postcss.config.js: Module exports like tailwind and autoprefixer

/tailwind.config.ts: Tailwind

/tsconfig.json: Typescript configuration

/README.md: Readme
```

Getting Started

To install all dependencies for the project, you can run:

```
npm install
```

To start the application:

```
cd api
```

for Mac users: `. .venv/bin/activate`

If this command results in an error, run `python3 -m venv .venv` first.

for windows users: `.venv\Scripts\activate`

If there are errors, 'pip install ...' dependencies

```
run python app.py
```

open a new terminal, cd frontend

```
npm install
```

```
npm run build
```

```
npm run dev
```

open the <http://localhost:3000> in a browser

Our backend:

By default, our app uses a backend hosted online. You can find `server/README.md` to learn more about our backend and how to run it locally instead.

How To Use Our Website

Home Page

You can view the trends page by clicking the "Home" tab on the navigation bar.

In the home page, at the top there is a navigation bar with tabs to you can toggle between "Home", "Build Schedule", and "Recommendation" as well as a light/dark mode, and a login/sign up section. The home page is formatted into three main sections, a "Watchlist", "Current Schedule", and "Classes Left to Take."

The Watchlist sections consists of randomly selected courses from a databaes of Computer Science and Mathematics courses. Four courses are randomly selected when the page is refreshed, each course is displayed in a container with teh title of the course, a brief description, and clicking the "View All" button allows a user to see more details about a course such as the Professor and the entire description. This section intends to give students reccomendations on courses to take, as well as relevant information to inform their decision.

The Current Schedule section is a table that intends to display the courses a user wants to put in their schedule. The Table ranges from Monday to Friday and times go from 8am to 7pm daily. This section intends to input the classes selected by the user into a table format to visualize their weekly schedule.

The Clases Left to Take section is a checklist of courses a student has left to take to complete their major. The check boxes consist of the core computer science requiremets a computer science student needs to complete in order to get their degree. Clicking the box next to a course will mark it as completed to allow students to easily see what courses they have taken.

The home page is the heart of our web application, and is intended to be the hub for where the user can quickly glance at relevant topics regarding course selection and schedule building.

Build Schedule

You can view the trends page by clicking the "Build Schedule" tab on the navigation bar.

In the build schedule page, at the top there is a navigation bar with tabs to you can toggle between "Home", "Build Schedule", and "Recommendation" as well as a light/dark mode, and a login/sign up section. The build schedule page is formatted into two main sections, a section that displays different courses and a list of courses selected for a semester.

The section that displays different courses is formatted into colorful side by side textblocks of potential courses labelled with their course IDs. These serve as an array of potential courses the student is considering, and allows a student to add them to their cart. Once added, a student can click generate to generate random schedules that follow the non overlapping schedule algorithm to create valid scedules. Moreover, clicking the calendar button in the right corner of a schedule will copy it to the home page. This section intends to allow students to have a broad overview of potential courses they could take.

The list of courses selected for the next semester is a panel to the right of the screen that dispalys the semester, and under it the courses a user has selected for an upcoming semseter, as wel as designations of whether it is a major requiremnt, or a general education requirement. The intention of this panel is to organize the courses a student plans to take, as well as how they distribute into their major requirements and university breadth requirements.

Recommendation

You can view the trends page by clicking the "Build Schedule" tab on the navigation bar.

In the build schedule page, at the top there is a navigation bar with tabs to you can toggle between "Home", "Build Schedule", and "Recommendation" as well as a light/dark mode, and a login/sign up section. The recommendation page is formatted into two main sections, a "Hot Courses" and "Student Feedback" section.

The Hot Courses sections consists of courses that are usually in demand courses from a databaes of Computer Science and Mathematics courses. Each course is displayed in a container with teh title of the course, a brief description, and clicking the "View All" button allows a user to see more details about a course such as the Professor and the entire description. This section intends to give students reccomendations on courses to take, as well as relevant information to inform their decision.

The Student Feedback section consists of shape containers with text in them meant to relay feedback from other students about courses. This section is under development as this requires backend integration from either Campuswire, RateMyTeacher, or another source. This section is intended to allow students to see what others have said about different courses to make more informed decisions while picking courses.

Bugs/Limitations

In it's current state, our scheudle buidler struggles with conflicting schedule times when generating schedules. Additioanlly, there is not databse for student comments, they have been harded coded in. Moreover, when geenrating schedules we had difficulty implementing a no overlapping times algorithm. Login and signup not fuctional.

Team Member Names:

Hanna Jiang

Joseph Schwab

Kailin Luo

Mary Foley

Soham Shah

Trung Quoc Nguyen

Vamsi Kontham

GitHub Respository Link:

Link: <https://github.com/cs320-mattsrats/schedule-builder>