Topic: Better course platform

Author: Sebastian Rodriguez Irizarry

Date: 1/27/2022

Problem Background:

- *At the moment the University of Puerto Rico is using an outdated platform for handling student admissions and registering their courses as well as the process of planning your next semester is more of a drag than helpful.*
- *The platform has recently had more downtime than ever before in crucial times where there's massive user flow as well as cause massive setbacks next semester.*
- *The frontend is outdated and doesn't streamline pre-enrollment nor course registration making students ultimately make an unfavorable selection of courses for next semester.*

Target

- *The goal is to create a scalable backend that would eliminate the current bottleneck, eliminate the need for queued enrollment days and eliminate student frustrations as well as have a good pre-enrollment dashboard that streamlines the process and makes for a better planned semester.*

Causes

- *Backend is not optimized for handling multiple hundreds of students at a time.*
- *Backend surely has an outdated schema and makes it hard to build good web apps with it.*
- *Its uses ssh for enrollment instead of a cloud infrastructure.*

Countermeasures

- *Make a scalable and robust backend to handle any number of students at any time, eliminating student enrollment days by their GPA and eliminating setbacks as well as build good relational data to streamline frontend development and build a robust web app.*
- *Have the backend be in a scalable cloud infrastructure where it has been battle-tested for these types of applications like Parse Server and eliminate semester setbacks*
- *Make a robust schema that allows relation data like Course: {courseCode: ‘INSO4101’, sections: [{section:’090’, professor: {name: ‘Marko Schütz-Schmuck’ , overview:’bla bla bla’}, etc…] for better front end features and maintainability.*
- *Use modern frameworks to save pre-enrollment options and follow material guidelines for a more robust and intuitive frontend eliminating previous student frustrations.*

Check/Evaluate

- Have a dedicated email for student feedback and any bugs they might encounter.
- Compare current metrics like downtime, setbacks, and user engagement in the site with the previous technology.

Act/Standardize

- *In the expected days where users will increase by the thousands, the server should increase its resources to realistically handle the traffic.*
- *Keep rolling out useful features on the enrollment website thanks to the robust schema that would be made in the backend.*
- *Encountered bugs will have priority as we are trying to eliminate previous student frustrations*
- *After some time polishing everything out, we could release some public endpoints for student innovation around the campus.*