Requirement 2:

1. I was confused on what to build at first. Therefore, it took me so much time to complete and send the project over. I was thinking of building a hotel reservation page. I started building that but then changed my mind and built this instead. My thought process was as follows: I created a trello board that had all the tasks I had to do. I broke down all my tasks into small pieces. I wanted to create the model first and implement the endpoints. I tested the endpoints with postman to make sure my requests work the way I want them to. After, I created a simple UI on the front end that would fetch the data and display it in a list fashion using the '.map' function. After I was confident that I could render the elements from the database, I tested out other requests. After I made sure all my endpoints are connected properly, I split up my project into different pages and set up a navigation to make it look and feel pleasant. Also, I displayed the data in a table fashion. I was thinking of what I should use for styling and I ended up using a CSS library called 'Bulma' that I am used to using for speed and simplicity. It is similar to 'bootstap' but it is easier to read. After I completed my project, I tried to clean up my code a bit by separating some components into a different file.
2. Write-up
    a. How I like to think about connection string is the string that you pass to a provider that passes it to a database. It is basically communicating between frontend and backend of the application. When you are fetching the information from the database, you do that using a specific string url that lets you get the information for that string. Imagine sending an email address. From your email to someone else's. You are able to pick that one person from the database to send the message to. As a result, you will have a private communication with that one person from the database. A similar situation happens in fetching a data. You can either pick one data from the database based on the id or you can pick multiple fields using a url that connects your frontend app with the backend api.

    b. I chose to use ORM for my application because I did not have an experience working with it prior to working on this project. Me and my other classmate were looking at it and wondering if it is a good idea to use it in this application. It was a good idea to go with ORM in Spring Boot because as students, we are familiar with java objects. It was easier to deal with classes as oppose to writing queries. I used Hibernate tool to handle the ORM for my application. One bit of code that from my application is:

        i. spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.PostgreSQLDialec

    This command lets hibernate generate better SQL queries for my database.

Requirement 3:

- Inspiration
    - I always wondered how the registrar's platform works. I wanted to know how they see students' status. As a result of this curiosity, I decided to build my

version of it and make sure it has somewhat of a good UI and UX. It makes the experience smooth and fast.