

## Comments

...

## User Stories: “As a user, I...”

- [ ] will land on a visually pleasing homepage by default, which allows navigation to view **all campuses** and **all students**
- can navigate to **all campuses view**, and
  - [ ] see a list of all campuses in the database
  - [ ] see an informative message if no campuses exist
  - [ ] add a new campus
    - [ ] with a validated form displaying real-time error messages
    - [ ] which redirects to the new campus' **single campus view**
- can navigate to a **single campus view**, and
  - [ ] see details about a single campus, including enrolled students (if any)
  - [ ] see an informative message if no students are enrolled at that campus
  - [ ] navigate to any student's **single student view**
  - [ ] delete the campus (and elegantly handle associated students)
  - [ ] edit campus information (including adding/removing students)
    - [ ] with a validated form displaying real-time error messages
    - [ ] which redirects back to the **single campus view**
- can navigate to **all students view**, and
  - [ ] see a list of all students in the database
  - [ ] see an informative message if no students exist
  - [ ] add a new student
    - [ ] with a validated form displaying real-time error messages
    - [ ] which redirects the new student's **single student view**
- can navigate to a **single student view**, and
  - [ ] see details about a single student, including the campus at which they are enrolled (if exists)
  - [ ] see an informative message if student is not enrolled at a campus
  - [ ] navigate to **single campus view** of the student's enrolled campus
  - [ ] delete the student
  - [ ] edit the student's information (including campus s/he is enrolled at)
    - [ ] with a validated form displaying real-time error messages
    - [ ] which redirects back to the **single student view**

Total: - / 25

## Rubric Score Legend

N/A	Not relevant to submitted code
0	Seriously needs improvement
1	Needs improvement
2	Satisfactory
3	Good
4	Excellent

### Frontend (Total: - / 8)

Criteria	Score	Comments
Does not misuse framework APIs (e.g. proper construction of promise chains with <code>axios</code> , does not mutate <code>react</code> or <code>redux</code> state, identifies use case for <code>state</code> vs. <code>props</code> , ...)	- / 4	
Integrates seamlessly with the backend and handles errors elegantly	- / 4	

### Backend Middleware (Total: - / 8)

Criteria	Score	Comments
Implements all routes in specs and connects to UI	- / 4	
Error handling middleware is present and appropriately utilized	- / 4	

### Models (Total: - / 12)

Criteria	Score	Comments
Model fields are correctly defined with appropriate validations	- / 4	
Associations are correctly made between models	- / 4	
Models are built with useful methods and configurations that utilize a wide range of Sequelize features (e.g. instance/class methods, hooks, getters/setters, etc...)	- / 4	

## Cleanliness and Maintainability (Total: - / 28 )

Criteria	Score	Comments
Avoids repetition in favor of reusable variables and functions (DRY)	- / 4	
No unused/unnecessary code blocks, variables, functions, ...	- / 4	
Uses meaningful variable names that are self-documenting	- / 4	
Formatting is consistent (whitespace, indentation, ...)	- / 4	
Does not contain blocks of commented-out code, except in form of documentation	- / 4	
Free of syntax errors and use of deprecated/unsafe features	- / 4	
Application follows a modular architecture and respects separation of concerns	- / 4	

## Extra Credit (up to 15%)

- Spectacularly styled and visually stunning (up to 5%)
- Additional features not specified in specs (up to 5%)
- Includes a functional seed file and script (up to 5%)

## Self sufficiency

- Help received outside of system issues, clarifications, or getting links to docs (percentage points subtracted on a case-by-case basis)

**Total:**        ( story points / 25 ) \* 60%  
                  + ( rubric points / 56 ) \* 40%  
                  + EC% - SS%

= \_\_\_\_\_%



