Name: Xinyu Huang
Email: xhuang40@jhu.edu
Section: 315
Name: Minwei Xu
Email: mxu31@jhu.edu
Section: 315

1. Overview

Project URL: http://university-frontend-server.herokuapp.com/

Code also available on Github: https://github.com/vanshady/University

Notice: Since both our frontend and backend servers are deployed on heroku, please allow a few minutes for both server to come up. Please refresh if the page keeps loading for more than 10s.

The US government released college scorecard data, a rich set of data on all US undergraduate degree-granting institutions. This is a project to query and display the analysis of the data.

On the top of the website, user can search for the university they want to know and information about completion rate, enrollment and degree will be displayed. Then you can see a statistical summary about all the university. User can choose different control, state, SAT score or race to see different summaries. In addition, and the end of the website, user can input 2 university names to inquire the distance.

2. Database Schema

We've made some changes to the database design to better reflect the relation between tables.

University	unitId	name	city_id	zip	url	main_campus	num_branches	control_id	historically_black	predominantly_black	men_only	women_only	alias	level_id	operating
	1	Johns Hopkins University	1	21218	www.jhu.edu	True	0	2	0	0	0	0	JHU	1	true

City	city_id	city_name	state_id	region_id	latitude	longitude
	1	Baltimore	24	2	39.32838	-76.62

State	state_id	state_name	postcode
	24	Maryland	MD

Control	control_id	detail
	1	public

Level	level_id	detail
	1	4-year

Admission	unit_id	admission_rate	vr25	vr50	vr75	mt25	mt50	mt75	wr25	wr50	wr75	sat_avg
	1	0.05	740	670	620	740	670	620	740	670	620	2100

Degree_Percentage	unit_id	degree_id	percentage
	1	1	0.02

Degree	degree_id	degree_name
	1	Agriculture

Enrollment	unit_id	percentage	type_id
	1	0.2	1

Tuition	unit_id	in_state	out_state	net	instructional_expenditures
	1	47060	47060	23199	77339

Completion	unit_id	less_than_4yr	type_id	percentage
	1	6	1	0.75

Student_Type	type_id	detail
	1	White

Student	unit_id	dependency	income_level	percentage
	1	independent	1	0.2

Income	income_level	detail	student_type
	1	$0-$30000	13

Debt	unit_id	data	type_id
	1	143310	11

Family	unitId	par_ed_hs	par_ed_ps	avg_inc	med_inc	avg_inc_dep	avg_inc_ind
		0.3	0.4	50000	50000	50000	50000

3. **Parse data**

We downloaded the MERGED2014_15_PP.csv on https://collegescorecard.ed.gov/data/ which is completely free. We then used Excel to split tables into 16 csvs, did some data cleansing and basic parsings. For more advanced parsing, we used Python3 and the code was put in parser folder. In the end, we use https://sqlizer.io/ to convert csv files to sql.

4. **Platform**

MySQL on ClearDB with a size of 1GB

5. **How to run code**

All SQL queries are in the SQL folder. Each table has a corresponding .sql file and procedures.sql contains all the procedures. Parsers are in the parser folder. Front-end code is in the frontend folder. Back-end code is in backend folder.The frontend and backend code as well as the database is already up on heroku. However, if you want to deploy locally:

1) To set up database: run 'mysql -u root load.sql'. It will automatically create a database called UNIVERSITY and import all the data and procedures
2) To deploy backend server locally: Run 'mvn clean install' and then 'heroku local web'.
3) To deploy frontend server locally: Run 'npm install -g yarn' to install yarn globally, run 'yarn' to install all the dependencies, and run 'npm start' to build and start the Node server.

6. **Specialized area: User Interface**

Data-driven, material style, and interactive interface.

7. **Analysis of limitations**

Data is outdated, and if there is new data, the parsing procedure needs to be automated.

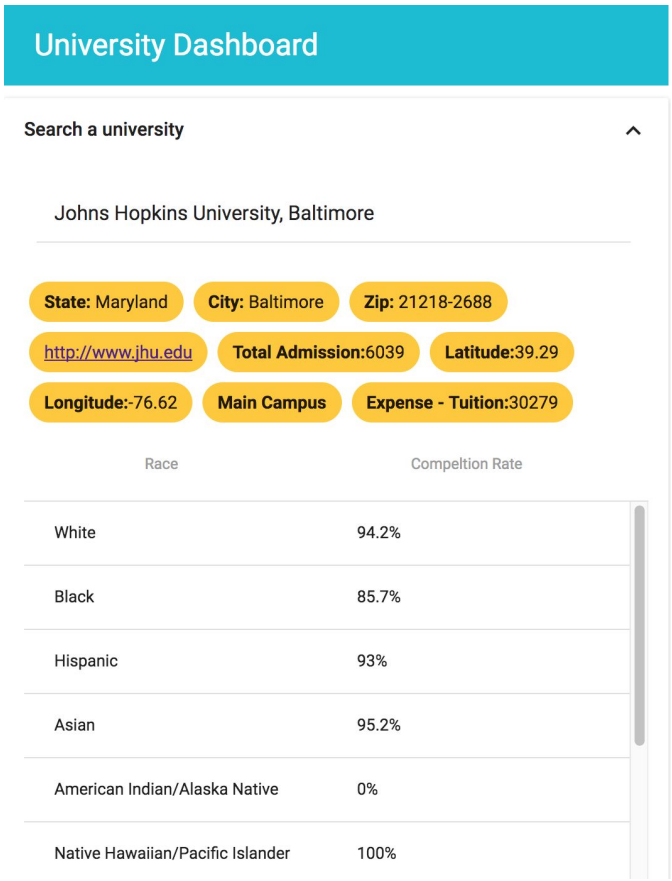
Querying speed is not ideal, need to use either memcached or server-side cache to restore the result since the data would not be updated.

Could use ORM to make the querying more easy to control.

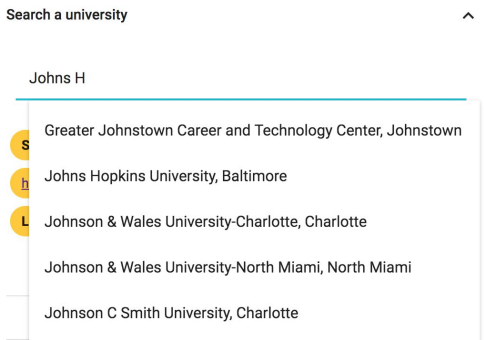
Could use GraphQL to make the requests easier and clearer.

Could use Redux or others to control the data flow in the frontend, so that there won't be too much cached data to drag the performance down.

8. **Output**



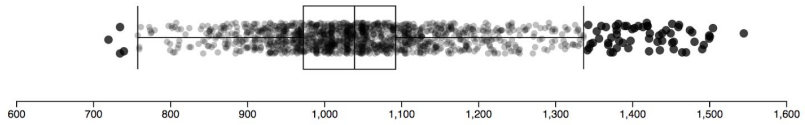
Query a university.



Autocomplete.

Below are the statistics for all universities

SAT Average Distribution



This is a boxplot showing the distribution of SAT average score of each university. The median average SAT score is 1039.5.

Data-driven boxplot of the distribution of SAT average grades.

Public University Tuition Difference

The average difference between out-state and in-state tuition of public universities is $6462.42.

Private University Tuition Expense Difference

The average expense - tutiion for private universities is $-13479.94.

Some statistics.

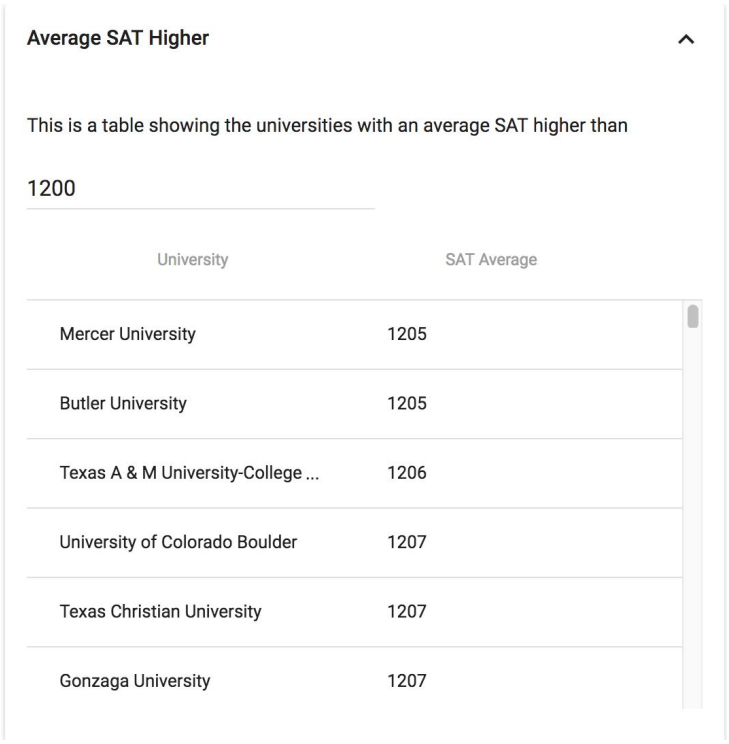
Average Enrollment Percentage

This is a table showing the average enrollment percentage per race for

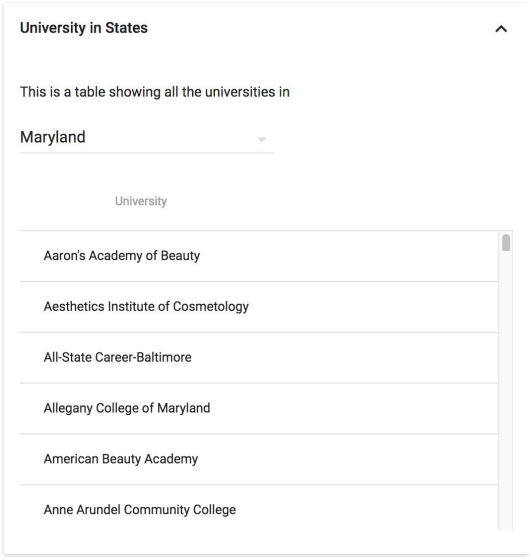
public universities
private non-profit universities
private for-profit universities

	Compeltion Rate
White	58.9%
Female	58%
Male	42%
Black	14.2%
Hispanic	13.5%
Asian	3.5%

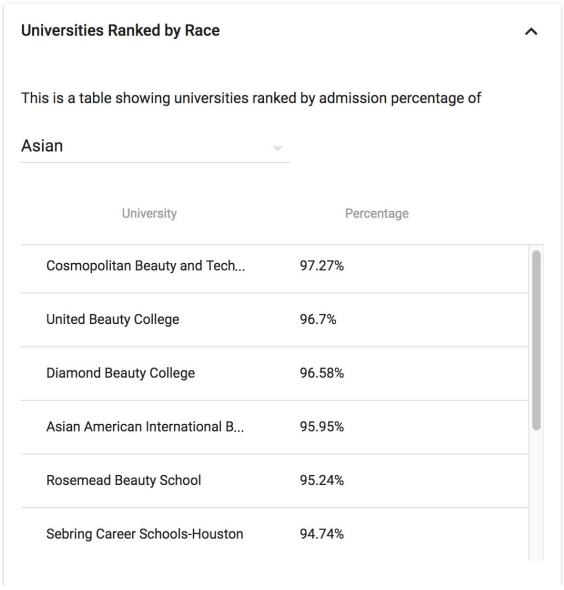
Average enrollment percentage per race for different type of universities.



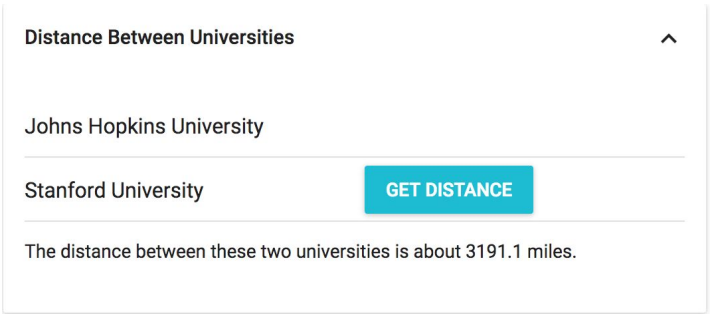
List all the universities with SAT higher than the given input.



List all the universities in a state.



Rank universities by admission rate for a specific race.



Compute distance between two universities.

9. Data Defining Language

```
CREATE TABLE Student_Type (
   `type_id` INT PRIMARY KEY,
   `detail` VARCHAR(33) CHARACTER SET utf8
);

CREATE TABLE Control (
   `control_id` INT PRIMARY KEY,
   `detail` VARCHAR(18) CHARACTER SET utf8
);

CREATE TABLE State (
   `state_id` INT PRIMARY KEY,
   `state_name` VARCHAR(30) CHARACTER SET utf8,
   `postcode` VARCHAR(2) CHARACTER SET utf8
);
```

```
CREATE TABLE Income (
   `income_level` INT PRIMARY KEY,
   `detail` VARCHAR(24) CHARACTER SET utf8,
   `student_type` INT,
   FOREIGN KEY(student_type) REFERENCES Student_Type(type_id)
);

CREATE TABLE City (
   `city_id` INT PRIMARY KEY,
   `city_name` VARCHAR(24) CHARACTER SET utf8,
   `state_id` INT,
   `region_id` INT,
   `latitude` NUMERIC(9, 7),
   `longitude` NUMERIC(10, 7),
   FOREIGN KEY(state_id) REFERENCES State(state_id)
);

CREATE TABLE Level (
   `level_id` INT PRIMARY KEY,
   `detail` VARCHAR(16) CHARACTER SET utf8
);

CREATE TABLE University (
   `unit_id` INT PRIMARY KEY,
   `name` VARCHAR(93) CHARACTER SET utf8,
   `city_id` INT,
   `zip` VARCHAR(10) CHARACTER SET utf8,
   `url` VARCHAR(124) CHARACTER SET utf8,
   `main_campus` BOOL,
   `num_branches` INT,
   `control_id` INT,
   `historically_black` BOOL,
   `predominantly_black` BOOL,
   `men_only` BOOL,
   `women_only` BOOL,
   `alias` VARCHAR(680) CHARACTER SET utf8,
   `level_id` INT,
   `operating` BOOL,
   FOREIGN KEY(city_id) REFERENCES City(city_id),
   FOREIGN KEY(control_id) REFERENCES Control(control_id),
   FOREIGN KEY(level_id) REFERENCES Level(level_id)
);
```

```
CREATE TABLE Admission (
   `unit_id` INT PRIMARY KEY,
   `admission_rate` NUMERIC(5, 4),
   `vr25` INT, `vr50` INT, `vr75` INT,`mt25` INT,`mt50` INT,`mt75` INT, `wr25` INT, `wr50` INT,
   `wr75` INT, `sat_avg` INT,
   FOREIGN KEY(unit_id) REFERENCES University(unit_id)
);

CREATE TABLE Completion (
   `unit_id` INT,
   `percentage` NUMERIC(5, 4),
   `type_id` INT,
   `less_than_4yr` INT,
   FOREIGN KEY(unit_id) REFERENCES University(unit_id)
);

CREATE TABLE Debt (
   `unit_id` INT,
   `data` NUMERIC(25, 19),
   `type_id` INT,
   FOREIGN KEY(unit_id) REFERENCES University(unit_id),
   FOREIGN KEY(type_id) REFERENCES Student_Type(type_id)
);

CREATE TABLE Degree (
   `degree_id` INT PRIMARY KEY,
   `degree_name` VARCHAR(80) CHARACTER SET utf8
);

CREATE TABLE Degree_Percentage (
   `unit_id` INT,
   `percentage` NUMERIC(5, 4),
   `degree_id` INT,
   FOREIGN KEY(unit_id) REFERENCES University(unit_id),
   FOREIGN KEY(degree_id) REFERENCES Degree(degree_id)
);

CREATE TABLE Enrollment (
   `unit_id` INT,
   `percentage` NUMERIC(10, 4),
   `type_id` INT,
   FOREIGN KEY(unit_id) REFERENCES University(unit_id),
```

```
    FOREIGN KEY(type_id) REFERENCES Student_Type(type_id)
);

CREATE TABLE Family (
    `unit_id` INT PRIMARY KEY,
    `par_ed_hs` NUMERIC(10, 9),
    `par_ed_ps` NUMERIC(10, 9),
    `avg_inc` NUMERIC(13, 7),
    `med_inc` NUMERIC(13, 7),
    `avg_inc_dep` NUMERIC(12, 6),
    `avg_inc_ind` NUMERIC(13, 8),
    FOREIGN KEY(unit_id) REFERENCES University(unit_id)
);

CREATE TABLE Student (
    `unit_id` INT,
    `percentage` NUMERIC(11, 10),
    `dependency` VARCHAR(11) CHARACTER SET utf8,
    `income_level` INT,
    FOREIGN KEY(unit_id) REFERENCES University(unit_id),
    FOREIGN KEY(income_level) REFERENCES Income(income_level)
);

CREATE TABLE Tuition (
    `unit_id` INT PRIMARY KEY,
    `in_state` INT,
    `out_state` INT,
    `net` INT,
    `instructional_expenditures` INT,
    FOREIGN KEY(unit_id) REFERENCES University(unit_id)
);
```

10. SQL Code

Please See load.sql, procedures.sql, and other sql code in the database folder.