INTRODUCTION
The goal for this project was to replicate the entire product development process from ideation, to design, to development of a fully-functional REST API, SQL Server that serves as a better alternative to RateMyProfessors.com. The website also attempts to embody all of the characteristics of a great website and incorporate industry-best practices.

WHAT MAKES A GOOD WEBSITE?
Characteristics:
1) Responsive
2) Fast
3) Great UX (User Experience)
4) Beautiful UI (User Interface)

PHASE 1: USER RESEARCH
Validate 4 Assumptions:
1) Majority of students use course review sites
2) RateMyProfessors.com is the most popular site
3) Majority of students have never written a review
4) Students are unable to find what they are looking for

All of the assumptions besides assumption #4 were validated through a user survey administered to college students. However, a separate finding revealed that:
Although students feel like they are able to find what they are looking for, over 95% of them expressed a need to view specific characteristics of a class.

PHASE 2: UI/UX DESIGN
During this phase:
- Created low-fidelity wireframes of what the website would look and feel like
- Conducted user interviews to find pain points to encourage user reviews (such as reducing # of clicks to submit a review)
- Designed high fidelity wireframes of final product

PHASE 3: DATABASE ARCHITECTURE
During this phase a MySQL database was designed and ensured each of the following:
- Each schema has a column to join with other tables
- Each table is minimal and only contained relevant columns
- Each schema is designed as industry-standard as possible, as opposed to putting all the data in one giant table

Over 16 schemas was created and also included identifying relationships.

PHASE 4: DEVELOPMENT
Stack: Nodejs, Express, MySQL (Backend)
EJS, ReactJS, HTML, CSS (Frontend)

Concepts:
- HTTP Requests to communicate with server
- Query string parameters used to identify data within DB
- JSON files to parse data from server to client
- Middleware to check login status before rendering pages
- MySQL prepared statements to prevent SQL injection
- Hashed passwords to ensure user privacy

Features:
- Login/Signup logic (web browser remembers if you logged in or not, password is hashed before being stored in DB)
- Fully dynamic website (pulls data from DB)
- React.js post requests (submitting ratings don’t require page refresh, updates client in real time)
- Occidental College restriction (only Oxy students can use it)
- Auto prompt for user ratings

ACKNOWLEDGEMENTS
- Celia Chen (Database design)