Rate My Class project final report

-- Team 29 AsWeCan

1. Briefly describe what the project accomplished.

RateMyClass allow users to lookup a specific CS course for its average GPA, user's ratings and user's comments. After logining in, users can rate a CS class in terms of interestingness, usefulness and difficulty, and give a comment to the class.

2. Discuss the usefulness of your project, i.e. what real problem you solved.

We believe that our users can gain much more information about a CS course than any other websites. Users not only care about the GPA of a specific section, but also care about how interesting a class is, how the difficult and how useful this class is. Through our website, users can lookup or rate a course from different perspectives

3. Discuss the data in your database

The GPA data of courses are from http://waf.cs.illinois.edu/ which is collected by professor Wade Fagen-Ulmschneider. Some of the users comments are crawled from the website "rate my professor". All other ratings are made by our users and are stored in our database.

4. Include your ER Diagram and Schema

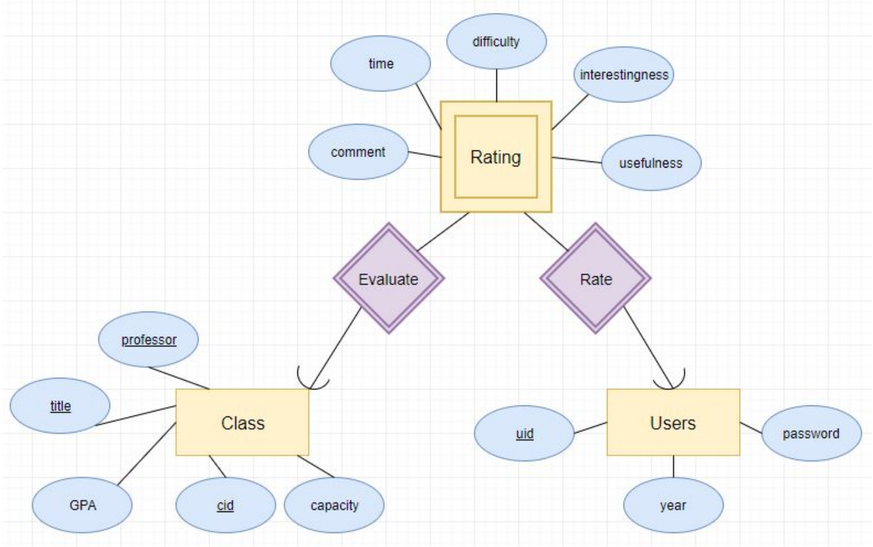
Schema:

Class(cid,professor,title,GPA,capacity)

Users(uid, password, year)

Rating(class.cid,class.professor,class.title, Users.uid,comment,time,difficulty, interestingness, usefulness)

ER Diagram:



5. Briefly discuss from where you collected data and how you did it (if crawling is automated, explain how and what tools were used)

The GPA data of courses are from http://waf.cs.illinois.edu/ which is collected by professor Wade Fagen-Ulmschneider. The original csv is provided in Professor Wade Fagen-Ulmschneider's Github.
Some of the comments of classes data are crawled from website ”Rate My Professor” by a Python program. We also do the survey to other students who have taken CS classes before and collect their reviews.

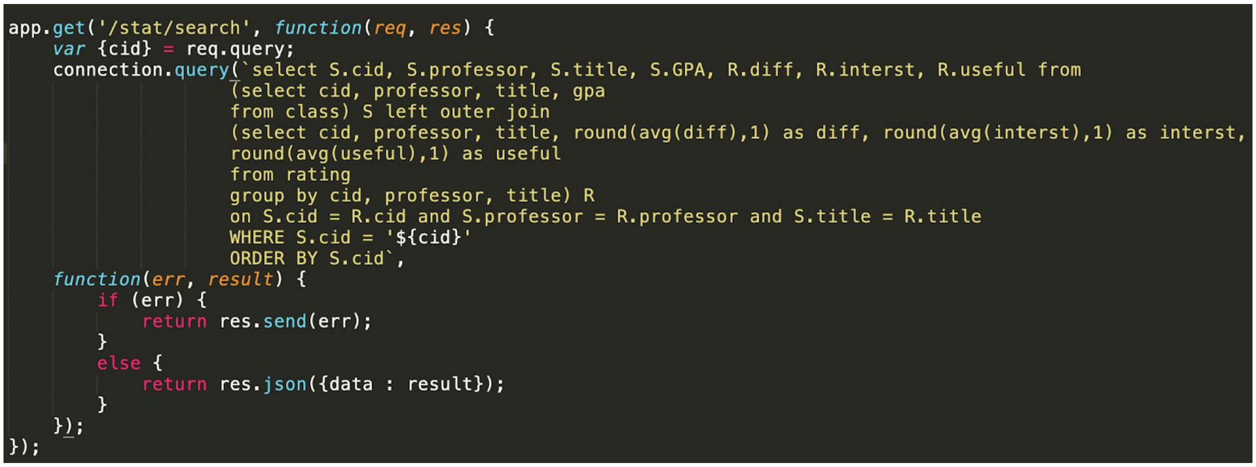
6. Clearly list the functionality of your application (feature specs)

a. Search course id
b. Insert, update, delete user reviews, including comments and 3 ratings (difficulty, interestingness and usefulness)
c. Login/Signup and show personal profile
d. Data visualization
e. Course recommendation based on machine learning prediction using comments
f. Generate tags based on comments

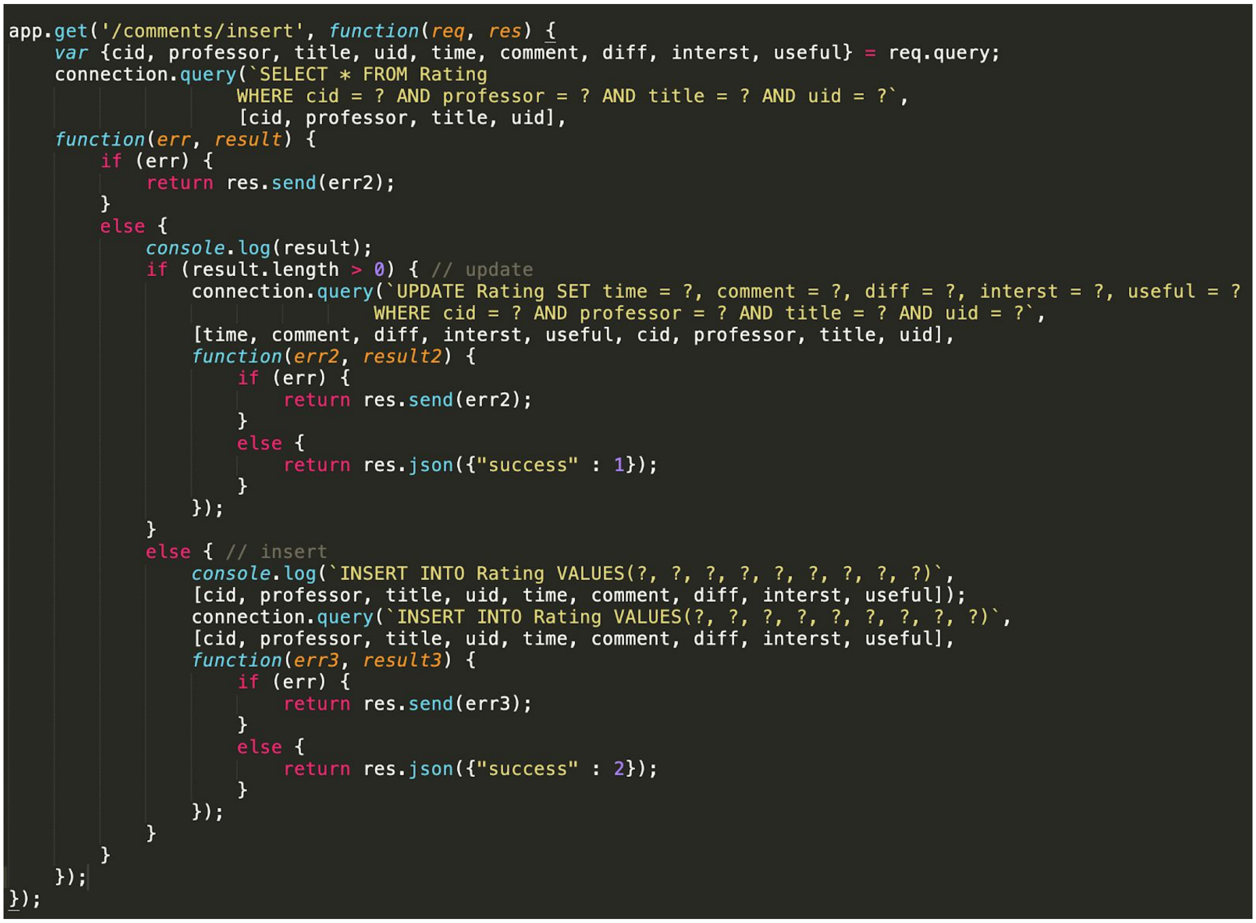
7. Explain one basic function

Search function:
By typing course number, users can search this course's different section and different professor, with average GPA and three rating index: difficulty, interestingness, usefulness. By clicking reviews, users can see all the comments and rating index in a new page. By clicking Rate, users can add new rating to this course.

8. Show the actual SQL code snippet



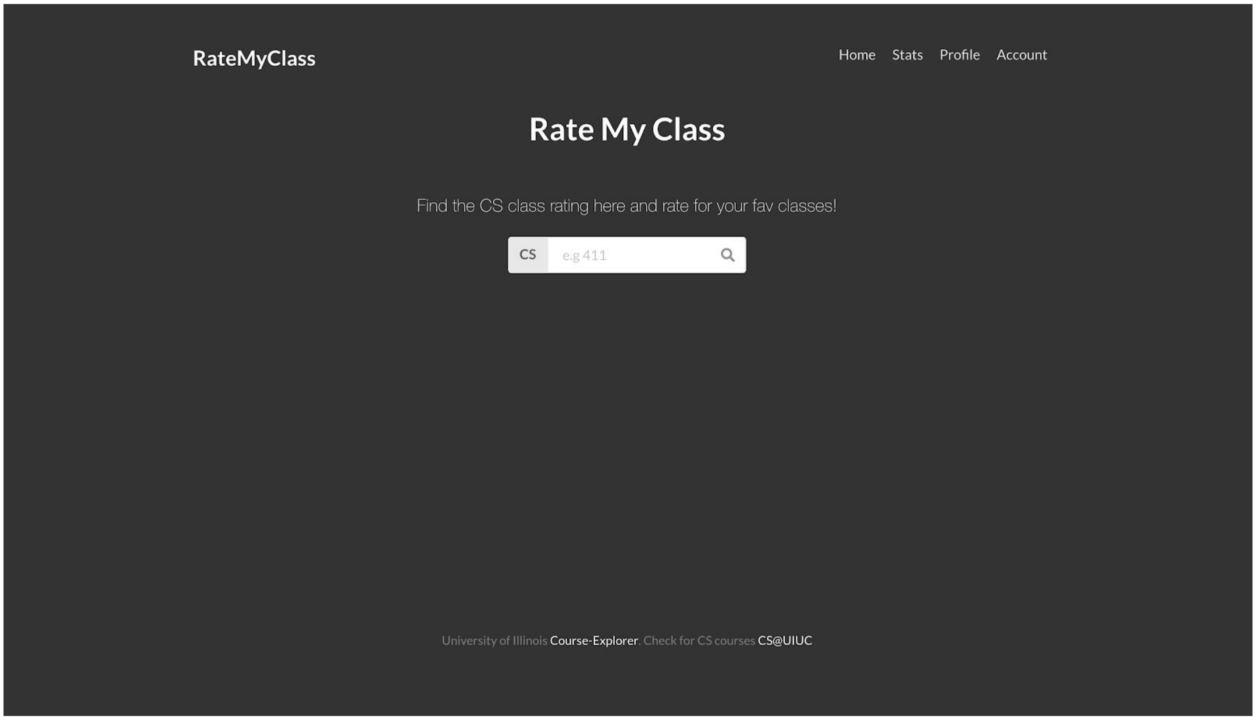
```
app.get('/stat/search', function(req, res) {
    var {cid} = req.query;
    connection.query(`select S.cid, S.professor, S.title, S.GPA, R.diff, R.interst, R.useful from
                    (select cid, professor, title, gpa
                    from class) S left outer join
                    (select cid, professor, title, round(avg(diff),1) as diff, round(avg(interst),1) as interst,
                    round(avg(useful),1) as useful
                    from rating
                    group by cid, professor, title) R
                    on S.cid = R.cid and S.professor = R.professor and S.title = R.title
                    WHERE S.cid = '${cid}'
                    ORDER BY S.cid`,
    function(err, result) {
        if (err) {
            return res.send(err);
        }
        else {
            return res.json({data : result});
        }
    });
});
```



```
app.get('/comments/insert', function(req, res) {
    var {cid, professor, title, uid, time, comment, diff, interst, useful} = req.query;
    connection.query(`SELECT * FROM Rating
                    WHERE cid = ? AND professor = ? AND title = ? AND uid = ?`,
                    [cid, professor, title, uid],
    function(err, result) {
        if (err) {
            return res.send(err2);
        }
        else {
            console.log(result);
            if (result.length > 0) { // update
                connection.query(`UPDATE Rating SET time = ?, comment = ?, diff = ?, interst = ?, useful = ?
                                WHERE cid = ? AND professor = ? AND title = ? AND uid = ?`,
                [time, comment, diff, interst, useful, cid, professor, title, uid],
                function(err2, result2) {
                    if (err) {
                        return res.send(err2);
                    }
                    else {
                        return res.json({"success" : 1});
                    }
                });
            }
            else { // insert
                console.log(`INSERT INTO Rating VALUES(?, ?, ?, ?, ?, ?, ?, ?, ?)`,
                [cid, professor, title, uid, time, comment, diff, interst, useful]);
                connection.query(`INSERT INTO Rating VALUES(?, ?, ?, ?, ?, ?, ?, ?, ?)`,
                [cid, professor, title, uid, time, comment, diff, interst, useful],
                function(err3, result3) {
                    if (err) {
                        return res.send(err3);
                    }
                    else {
                        return res.json({"success" : 2});
                    }
                });
            }
        }
    });
});
```

9. List and briefly explain the dataflow, i.e. the steps that occur between a user entering the data on the screen and the output that occurs (you can insert a set of screenshots)

1)Home page:

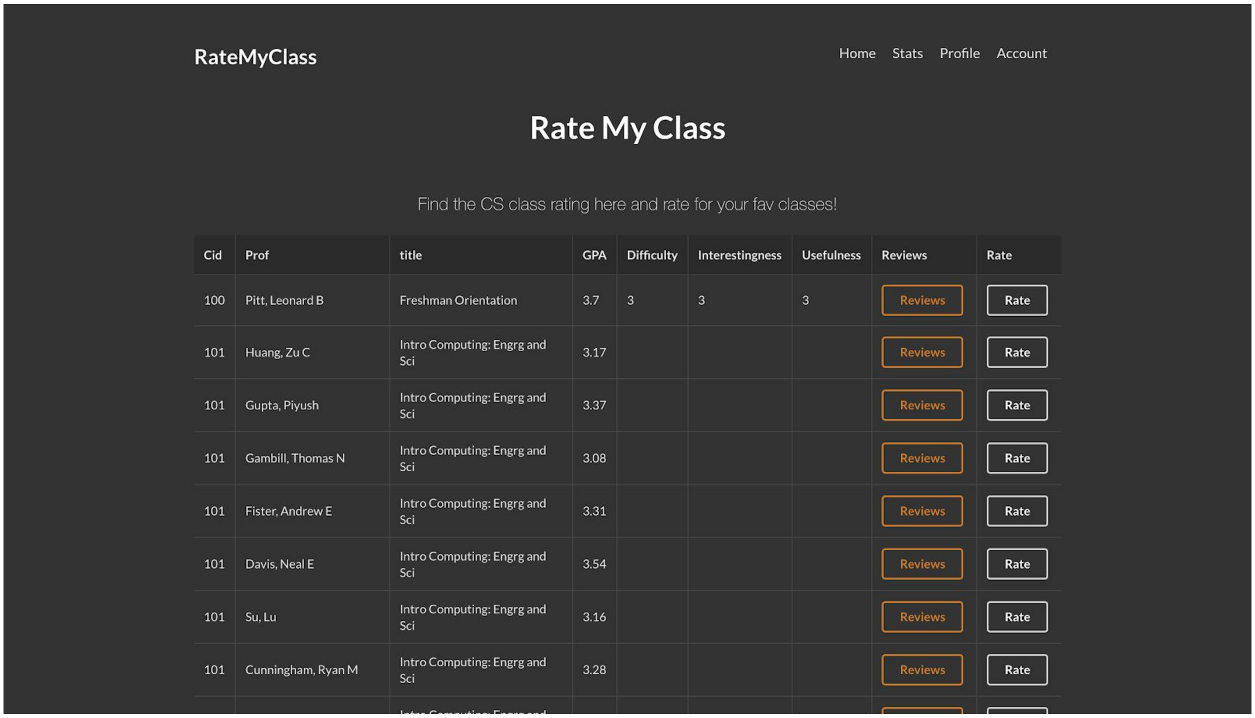


Rate My Class

Find the CS class rating here and rate for your fav classes!

University of Illinois Course-Explorer. Check for CS courses CS@UIUC

2)Stat page(with data of all the sections of CS courses):

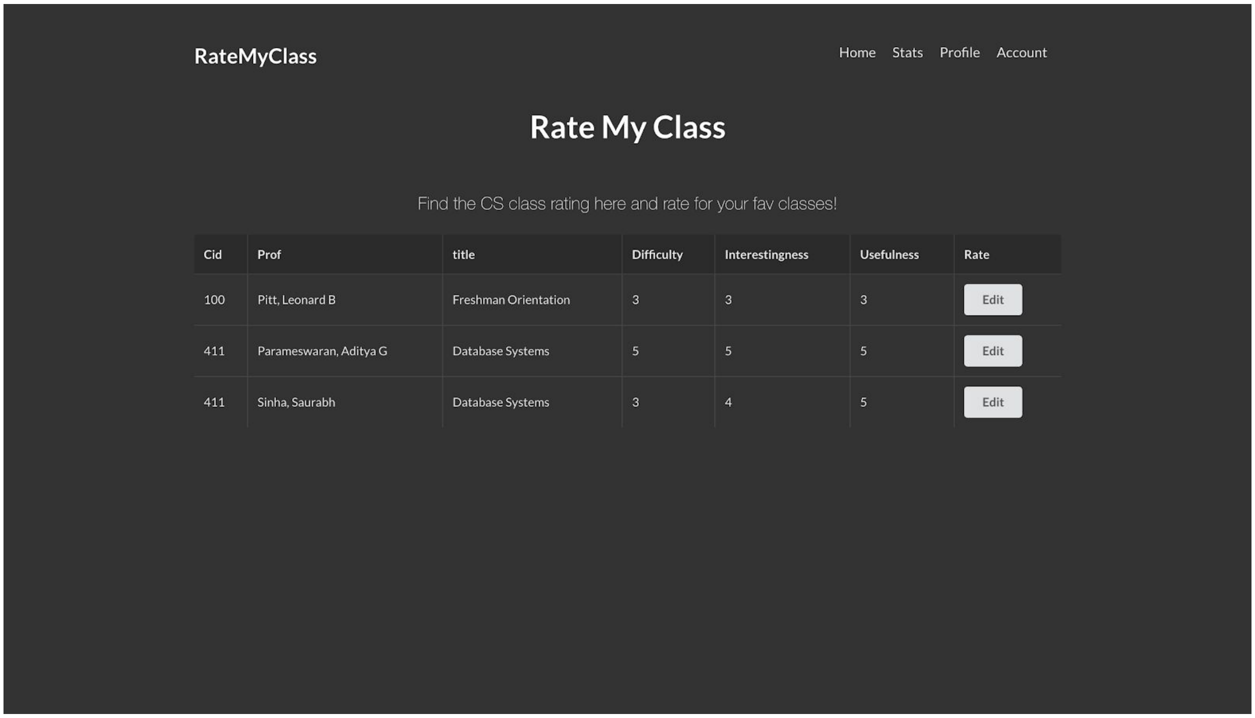


Rate My Class

Find the CS class rating here and rate for your fav classes!

Cid	Prof	title	GPA	Difficulty	Interestingness	Usefulness	Reviews	Rate
100	Pitt, Leonard B	Freshman Orientation	3.7	3	3	3	Reviews	Rate
101	Huang, Zu C	Intro Computing: Engrg and Sci	3.17				Reviews	Rate
101	Gupta, Piyush	Intro Computing: Engrg and Sci	3.37				Reviews	Rate
101	Gambill, Thomas N	Intro Computing: Engrg and Sci	3.08				Reviews	Rate
101	Fister, Andrew E	Intro Computing: Engrg and Sci	3.31				Reviews	Rate
101	Davis, Neal E	Intro Computing: Engrg and Sci	3.54				Reviews	Rate
101	Su, Lu	Intro Computing: Engrg and Sci	3.16				Reviews	Rate
101	Cunningham, Ryan M	Intro Computing: Engrg and Sci	3.28				Reviews	Rate

3)Profile page(all the rating from the user

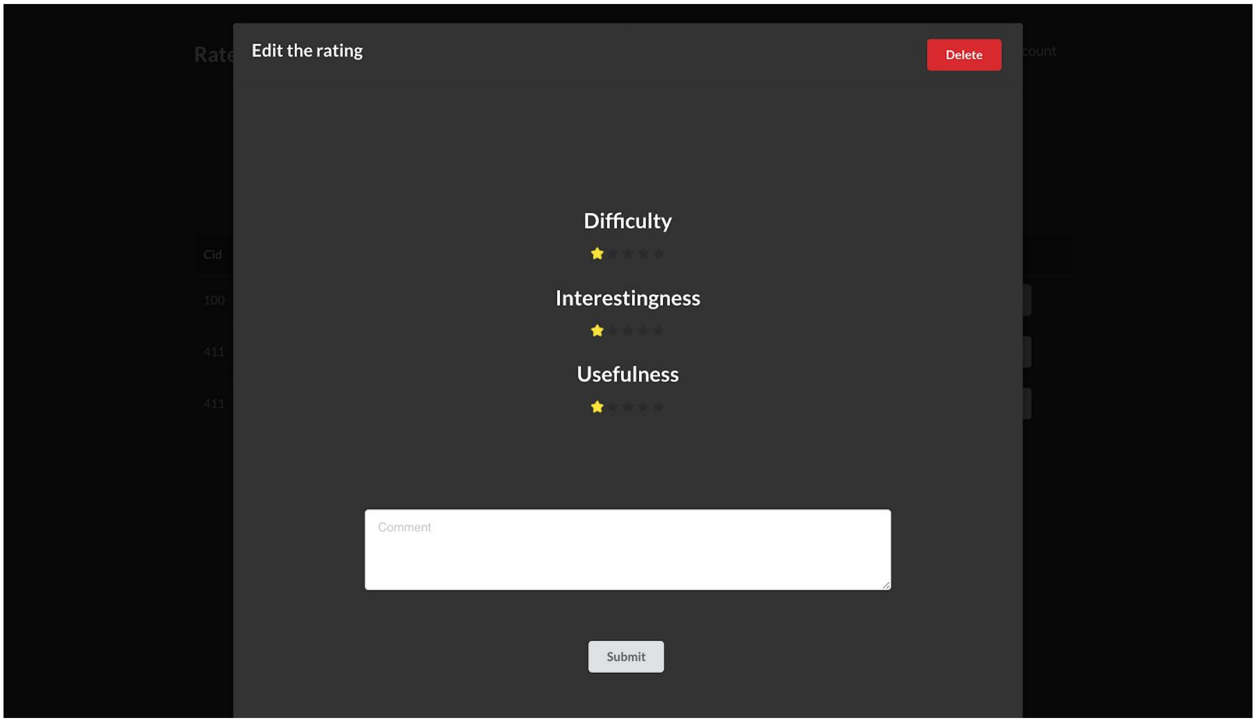


Rate My Class

Find the CS class rating here and rate for your fav classes!

Cid	Prof	title	Difficulty	Interestingness	Usefulness	Rate
100	Pitt, Leonard B	Freshman Orientation	3	3	3	Edit
411	Parameswaran, Aditya G	Database Systems	5	5	5	Edit
411	Sinha, Saurabh	Database Systems	3	4	5	Edit

4) Edit user’s rating: the users can ***edit/modify*** *the rating they rated before or* ***delete*** *the rating.*



Edit the rating

Difficulty

★☆☆☆☆

Interestingness

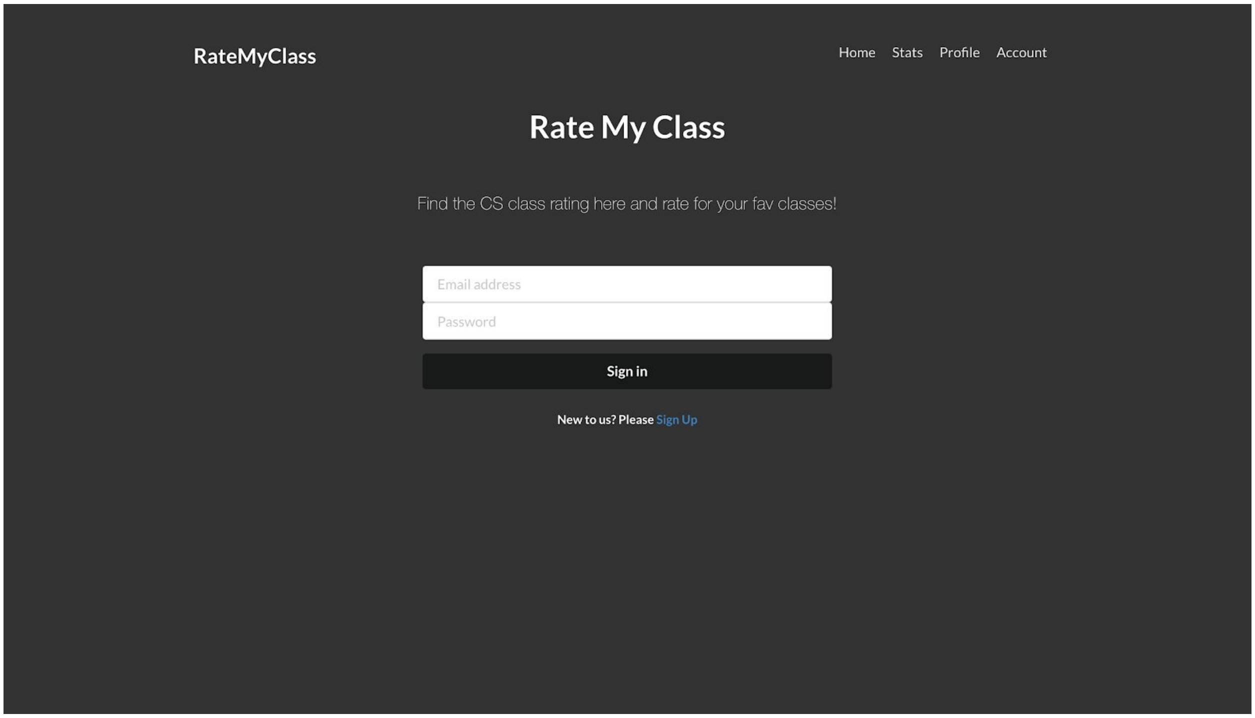
★☆☆☆☆

Usefulness

★☆☆☆☆

Comment

5)Login page(sign up and sign in)

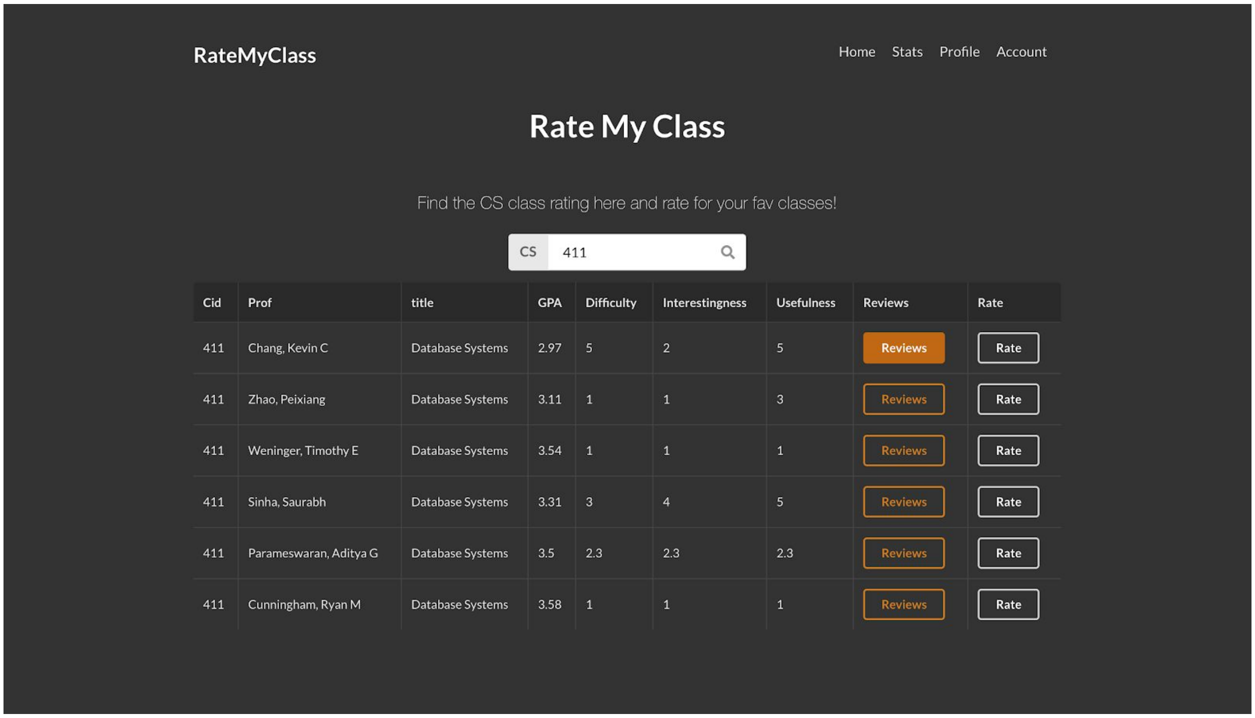


Rate My Class

Find the CS class rating here and rate for your fav classes!

New to us? Please Sign Up

6)Search page(search by class number)



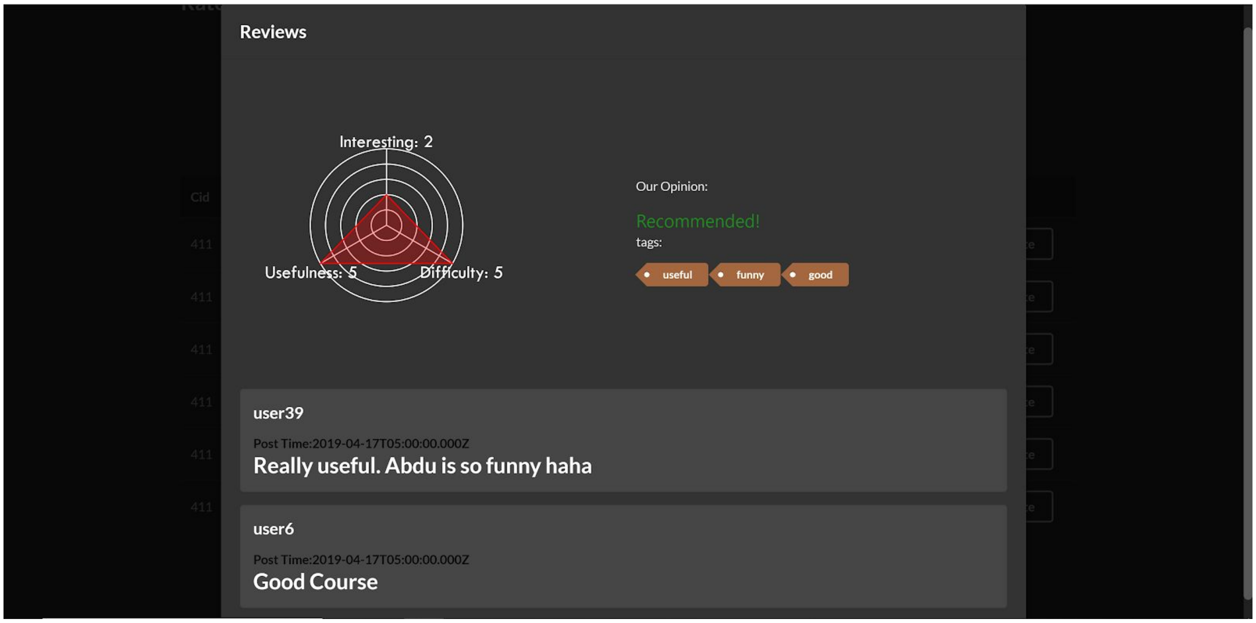
Rate My Class

Find the CS class rating here and rate for your fav classes!

CS 411

Cid	Prof	title	GPA	Difficulty	Interestingness	Usefulness	Reviews	Rate
411	Chang, Kevin C	Database Systems	2.97	5	2	5	Reviews	Rate
411	Zhao, Peixiang	Database Systems	3.11	1	1	3	Reviews	Rate
411	Weninger, Timothy E	Database Systems	3.54	1	1	1	Reviews	Rate
411	Sinha, Saurabh	Database Systems	3.31	3	4	5	Reviews	Rate
411	Parameswaran, Aditya G	Database Systems	3.5	2.3	2.3	2.3	Reviews	Rate
411	Cunningham, Ryan M	Database Systems	3.58	1	1	1	Reviews	Rate

7) Review for a class:



Reviews

Our Opinion:

Recommended!

tags:

user39

Post Time:2019-04-17T05:00:00.000Z

Really useful. Abdu is so funny haha

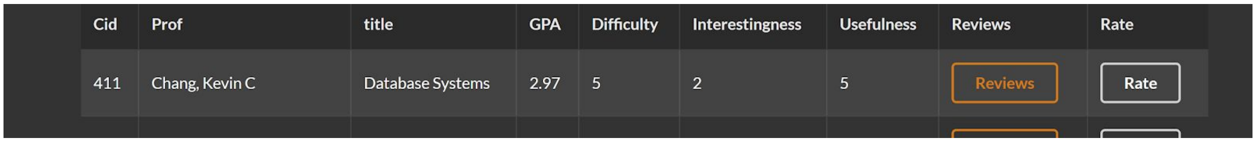
user6

Post Time:2019-04-17T05:00:00.000Z

Good Course

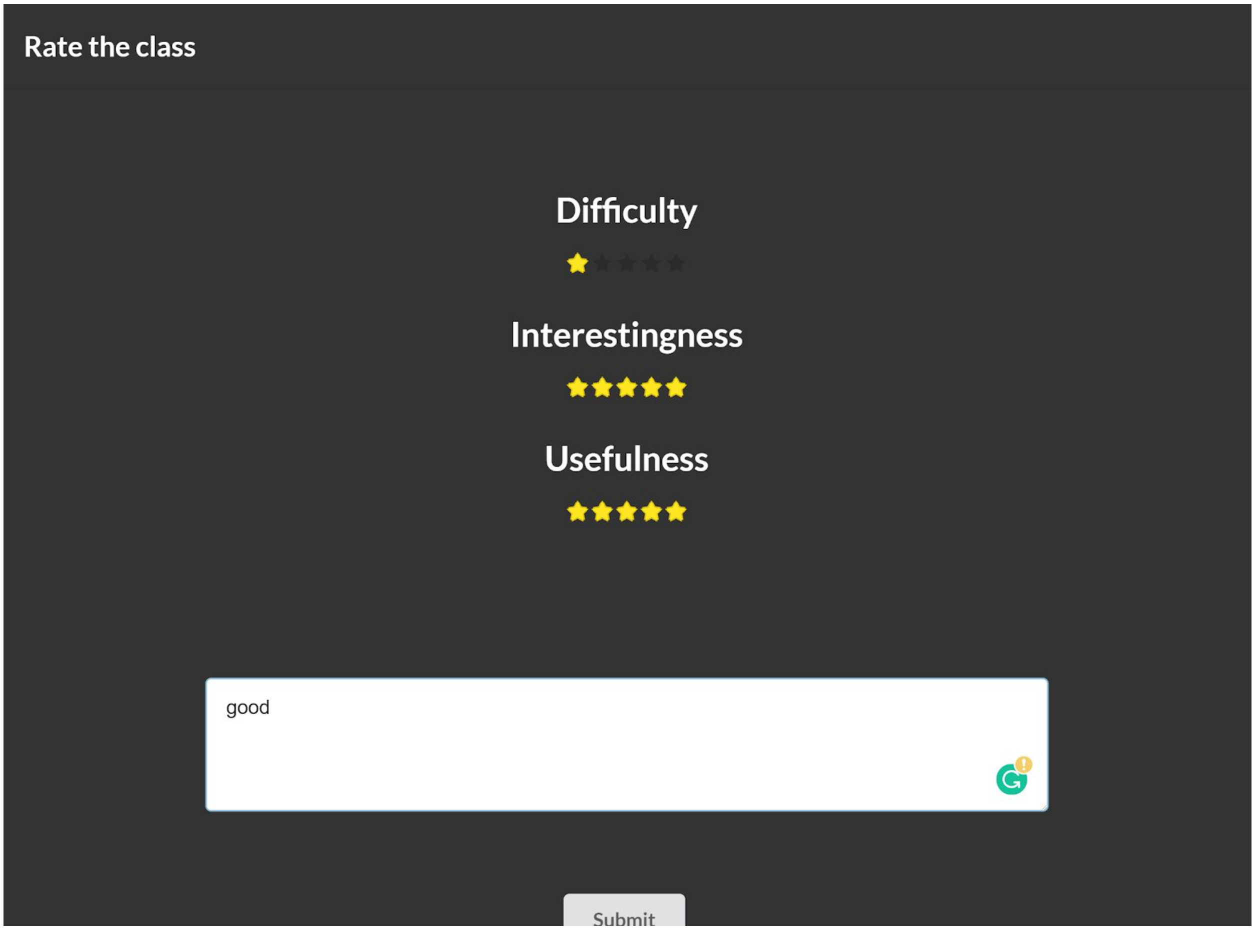
8) Rating for a class (use CS411 as example)

The example class looks like this



Cid	Prof	title	GPA	Difficulty	Interestingness	Usefulness	Reviews	Rate
411	Chang, Kevin C	Database Systems	2.97	5	2	5	Reviews	Rate

Insert a rating



Rate the class

Difficulty

★☆☆☆☆

Interestingness

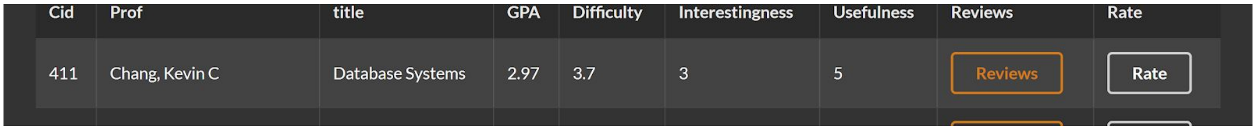
★★★★★

Usefulness

★★★★★

good

Then the class would be like following with changed values



Cid	Prof	title	GPA	Difficulty	Interestingness	Usefulness	Reviews	Rate
411	Chang, Kevin C	Database Systems	2.97	3.7	3	5	Reviews	Rate

10. **Explain your two advanced functions and why they are considered as advanced. Being able to do it is very important both in the report and final presentation.**

    a. Data visualization:

        1. Plain numbers are boring. We want to our users can have a clearer and better understanding of the shown data, so we implemented a radar chart, which presents the data of our three attributes(interestingness, difficulty, usefulness) in a 2D way. When data is inputted by another user, the chart will dynamically reflect the changes after a page refresh. We came up with the design that is clearest and nicest.
        2. Sometimes there are too many comments for a course. We want to do a auto text processing of all comments of a section and select the top content(meaningful) adjective words as tags. Pos-tagger(a natural language

processing tool to distinguish noun, verbs and so on) is not perfect so we need to use data mining technique to filter out those useless words which are not appropriate to be tags.

b. Machine Learning: We collected training data in two ways: one is using crawler to get comments from "https://www.ratemyprofessors.com/", and the other is by sending Google forms to other students in CS department. With these data, we implemented bag-of-words to encode sentences into vectors, with appropriate choice of stop words. Then, we tried two ways of classifying data, including nearest neighbor and logistic regression, and we chose the latter. Due to small training data size (less than 300), of which most are positive reviews, we found the recommendation prediction highly unreliable. For that reason, we chose logistic regression model, so that we could adjust the criteria by typing in different inputs. Finally, by fitting comments into bag-of-words and our model, we can make relatively accurate prediction about whether this course is recommended by most users.

11. **Describe one technical challenge that the team encountered. This should be sufficiently detailed such that another future team could use this as helpful advice if they were to start a similar project or were to maintain your project. Say you created a very robust crawler - share your knowledge. You learnt how to visualize a graph and make an interactive interface for it - teach all! Know how to minimize time building a mobile app - describe!**

One of the most important problem we have faced is how to temporarily store data in the front end. In the sign in design part, we need the front-end to remember if we have logged in, and the user id we are now currently using to access our database. Since we use React as our front end design library, we searched for tutorials in the Internet. The Net Ninja, a youtuber who mainly post some web development tutorial, really help use a lot. We use the Redux reducer as the way to solve the question. The idea is that we would design some data reducer, which can store the data change from components to the Redux Store, and apply back to the component when needed.

Here is the link to the youtube playlist: https://www.youtube.com/watch?v=Oi4v5uxTY5o&list=PL4cUxeGkcC9iWstfXntcj8f-dFZ4UtlN3, they introduced how to combine React, Reduce and Firebox database to design a database-based web app.

12. **State if everything went according to the initial development plan and proposed specifications, if not - why?!**

Our work went according to plan, and we finished most of the functions we planned to accomplish.

As for the different parts, one thing is we made minor adjustments on database. Some tables have useless columns, while others need extra primary keys. The other is we originally plan to group all courses with the same course id together (for example all cs 411 sections). However, we finally find the results not as useful as expected, since ratings vary from section to section. Some professors have much higher GPA and ratings than average. Besides, courses like cs 498 have different topics for each section. Therefore, we gave up this function. We also noticed a bug right before final demo that we cannot show our advanced functions, but we included them in the final video.

13. Describe the final division of labor and how did you manage team work.

Junquan Chen: Front-end development and data visualization advanced function
Yanjun Guo: Back-end development, machine learning advanced function
Wenhao Gu: Comment data crawler and website release
Zhesong Wu: Database System and Front-end development