

Project Reflection Report

Insightify is an application that collects and assembles data at assignment level granularity for different courses offered at UIUC every term and also shows reviews for courses and assessments. This application displays data at different levels of a course - from assignments, and exams to the entire aggregated course level. Each student can also review the course at an assignment level and provide his/her review with multiple points. Students can check this tool to decide whether to pursue a course in a particular term or not depending on their analysis of the data points.

1. Please list out changes in directions of your project if the final project is different from your original proposal (based on your stage 1 proposal submission).
 - a. Our final application has all the functionalities proposed in the initial proposal except for one part: we initially wanted to provide the instructor point of view as well, i.e instructor can access some student performance related data points from the application. But, this is not implemented in the final application.
2. Discuss what you think your application achieved or failed to achieve regarding its usefulness.
 - a. Our application achieved all the functionalities such as providing difficulty, workload, average rating mentioned in the original proposal successfully.
 - b. We have successfully implemented the application to display overall reviews of the course along with individual workload, review, technologies involved in the assessments of the course.
 - c. We could not implement the instructor centric part of the application because it was difficult to obtain submission history of individual students for each assessment in each course.
3. Discuss if you changed the schema or source of the data for your application
 - a. We implemented the database using the [UIUC dataset](#) as we wanted in the initial phases. The only minor change we made is that we did not add the “CourseURL” column in the Course table and the DepartmentURL column in the Department table due to lack of time.
4. Discuss what you change to your ER diagram and/or your table implementations. What are some differences between the original design and the final design? Why? What do you think is a more suitable design?
 - a. We did not make any changes to the ER design we made. We implemented the database as it is in the ER diagram.
5. Discuss what functionalities you added or removed. Why?

- a. The functionality added to the original proposal is to display the average GPA of a course for each semester along with the instructor name, average workload and average rating.
 - b. We did not implement the median, minimum and maximum, percentiles of gpas as we felt that it would supersede the important information.
 - c. We have added the statistics (grade along with count of students received that grade) for each instructor in a specific term for a specific course selected.
6. Explain how you think your advanced database programs complement your application.
 - a. We used triggers and stored procedures in our application. These advanced database programs have been very useful:
 - i. The triggers we added helped us to identify missing course-term entries in the course schedule pages
 - ii. We also used triggers to update and scale the user inputs for reviews
 - iii. Stored procedures helped us to integrate our advanced queries into the application and helped us find the difficulty tags based on ratings.
7. Each team member should 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 where to maintain your project.
 - a. Bala Sukesh (bbelide2):
 One challenge we faced is the database connectivity from Flask. We made some errors while writing database queries and executing them inside the flask code - one example is that we did not understand when to connect, commit and disconnect the database which caused errors and inconsistencies during executing database queries. We resolved this by connecting, committing and disconnecting inside each flask method before and after each use.
 - b. Ruthwik(ruthwik2):
 Another challenge which we faced is passing the values of a variable from one page to another page or to the backend python file. We figured out how to send the values from the HTML page to the backend using the route. Another challenge faced is implementing and integrating triggers in the python code.
 - c. Sri Lekha (sk121):
 The Database is designed at a very high level. The Enrollments table and reviews table have data entries which are more than 300,000 each which made some difficulties in maintaining the scalability of the database during the process of building our application as most of our data is autogenerated.
 - d. Satvik(satvikp3):
 Another challenge we faced was the lack of datasets we could use for our project. For data regarding courses, instructors and departments at UIUC we were able to find data but for all the other tables the data had to be fabricated, starting from

students info right down to assessment level data. We had to use libraries like faker and mix and match some datasets from kaggle to make up for that.

8. Are there other things that changed comparing the final application with the original proposal?
 - a. No
9. Describe future work that you think, other than the interface, that the application can improve on
 - a. We can improve the application with the following new features and functionalities:
 - i. More statistics other than averages
 - ii. More control to the user - adding filters to select courses, terms, instructors etc and getting the data only for the filtered results
 - iii. Better visualization - graphs, pie charts etc
 - iv. Adding shareable links for reviews
 - v. Adding external links - like links to course pages, university pages, faculty pages etc wherever required
 - vi. Cloud hosting
 - vii. Separating front and back ends for scalability
 - viii. Making the application more robust by adding security, authentications, and scalability with load balancers etc
10. Describe the final division of labor and how well you managed teamwork.
 - a. For every stage, all four of us collaborated and contributed equally. During the initial stages of project description documentation, database design and implementation, we all worked together discussing the use case, database design and coming up with ideas. During the implementation phase too we all collaborated together on all the parts, but on high-level the following division was followed:
 - i. Database implementation and data collection - Satvik and Sri Lekha
 - ii. Front End - HTML pages - Sri Lekha
 - iii. Flask code implementation - Ruthwik and Bala Sukesh
 - iv. ML integration for detecting Text review profanity and gibberish reviews: Satvik