We built this application due to a real need three of us face. **The lack of reviews on modules**. We went around to ask our friends how did they do their due diligence on a module before deciding to take it.

Peers are the most frequently cited source of course information, exceeding even the official catalogue information on IVLE and NUSMods and far surpassing the number of mentions of the university's faculty and professional advisors. We also asked "Are there any other sources of information you rely on when selecting courses?" The responses to this question were mostly "no," but there was a long tail of answers that included, "upperclassmen,", "personal preference," "old syllabus," "office hours," "previous course offering websites,""professor biographies," and "mostly what fits into my schedule."

It could be argued that most people will just ask their peers. But the number of reviews you could get from peers is bad from a statistical standpoint, too small of a size. Also, what if you are looking to take electives outside of your faculty?

- People can only choose among options of which they are aware, hence, it severely restricts students the opportunity to take modules and learn beyond their social circle.

- Students tend to select courses based on partial, poorly integrated information, and are typically restricted by logistical, informational, and normative constraints that have little directly to do with academics (Nathan 2005; Rosenbaum, Deil-Amen, and Person 2006)

- Many have critiqued the elective model, pointing out its costs and risks to students who navigate a complex academic landscape with little guidance (Goldrick-Rab 2006; Rosenbaum et al. 2006; Bailey et al 2015)

We feel the benefits for having a plethora of reviews to be:

- Encourages students to be more holistic by allowing students to get more information on modules outside their core majors where they may not know other undergraduates to consult on how the module is really like.

- Allows students to better select the courses, by resolving partial and poorly-integrated information.

- Mitigate the costs and risks to students and allow them to better plan their curriculum, be it for good grades, learning, exploring etc.

- Could give rise to new insights for the teaching staff since people may review differently when the data is made publicly even though it is anonymous

A common question posted to us would be why not just work on NUS mods, where they already have a review system in place. Till date, NUS mods has little to no reviews for most modules despite having that feature for a period of time.
This is something we wish to tackle but we felt developing on NUS mods would limit our ability to solve the issue. We thought about it and went around talking to our peers.

We came to the conclusion that NUS mods lack reviews due to:

- The culture in Singapore does not encourage people to leave reviews
- NUS mods review system may be too standard
- People are in general lazy to leave reviews unless they are forced or incentivised (etc. glassdoor)

We decided to tackle the problem based on these assumptions. We aim to make our application fun and engaging so it may circumvent these issues and organically encourage people to post reviews. Also, professors in NUS constantly change the modules they teach, hence, we feel that perhaps a professor review system may be more accurate than a module review system.

We designed our application with the aim to make it fun, making the icons friendly and the reviews look more entertaining. When we told our friends about this application and gave them the link to view it, they found it entertaining and some mentioned that they may post reviews on some of the professors.

Apart from the benefits we could potentially bring to our end users, we feel we have **engineered** a decent product in such a short span of time.

- We managed to achieve a full stack architecture despite having only 3 members, with the frontend, backend and database on seperate servers.

- A restful API that could be ported on and run on any server easily.

- A client side that could be ported on and run on any server easily.

- Application could go live anytime

Things we could have improved could perhaps be introducing https to both client and API side. We decided that we did not need it since we are not dealing with critical information, credit cards etc. Hence, a man in the middle attack may not be our greatest concern and we rather push out other features or better the user experience in terms of design and workflows. However, this decision has cost us as we are unable to utilise service workers which decreased our offline functionality. We could also improve on caching on the client side to reduce load on the API side.