# FAQ

1. A course-offering (e.g., COMP1531 17s2) will have only survey form associated with it." mean?

A course offering here refers to a course with a session associated with it. So, COMP1531 is a course and one possible course offering would be COMP1531 17s2. The guidelines indicated that one course offering can only have one survey form associated with it. So, COPMP1531 17s2 can only have one survey form and COMP1531 18s2 can only have one survey form and so on.

2. Files must be loaded into the database by the survey system when the application is started." Is this right? If the courses change it would likely corrupt/make some surveys pointless since we have to store what course the survey belongs to

Courses already in the file will not change, hence surveys associated with these courses will not be impacted. Changes to the file will constitute any new courses being added, then the database will be accordingly updated to add the new courses

3. How do we take care of changes to “csv” files? What if there are duplicates?

Typically, your application would check if the data to be inserted already exists in the table before every insertion. So, each time your application is started, all the data is not inserted again, only the new data. Again, in a typical production environment, when you are relying on external data (such as the csv files in our project), there could be errors in the data (e.g., duplicates). Your application will have to safeguard against this kind of data.

4. Can we implement the application using a different technology stack instead of Python/Flask/Jinja2? Can we used a different database e.g., postgresql

This application must be implemented using the technology stack (Python/Flask/Jinja2), so that we can have a consistent marking scheme for all students. However, you can choose to use a different database (e.g., Postgresql), though we prefer you to use SQLite3. If you choose to use a different database, the tutors will not be able to offer technical assistance. We still need to see your schema layout and SQL scripts.

5. How do we implement the database tier? Should we have SQL scripts to create tables?

There are a few different ways you can choose to implement the database tier.

(i) Use SQL scripts (run on a SQL shell) to create all the tables, create the admin credentials etc. If you choose to do this and use Python application code for data insertion/querying only, this is fine. But we will need to see the SQL script that was used to create the table definitions.

(ii) Using Python with raw SQL or SQLAlchemy (week 10 lecture) for creating and maintaining the data tier.

In both approaches, make sure when you start the application, your application takes care to only insert new records and has checks for duplicate records or any primary key violations.

6. How many surveys should admins be allowed to create per course offering?

Refer to answer in question 1. As each course-offering can only have one survey, an admin should only be able to create per course offering?

7. If our team is using a template for the design of the system, how do we credit the source?

Firstly, note that you are permitted to use a template only for the front UI of the application. The back-end has to be your own implementation. To credit the source, in your final report, credit the source (e.g. the URL where you downloaded the sourc-code) on which your UI design is based.

### 8. How should the answers for text based questions appear in the result page?

A screen-shot from myExperience is shown below that shows one possible way of displaying text-based results.


### 9. Do the students and staff members get access to a graphical display of survey results or is this just the admin?

Students and staff will be able to get access to a graphical display of survey results only after the survey has closed. You might choose to generate a graphical display of results for all users after the survey has closed. Alternatively, admin might be able to generate a graphical display of results at any stage throughout the life-time of the survey. The latter is probably a more realistic design similar to our demo application.

### 10. If there are multiple reviewers (staff) for a course, then when is the survey open to students to fill out?

There appears to be a few variations in implementing this:

(i) Survey appears on the dashboard of all staff associated with the course offering, but as soon as one staff reviews it, it disappears from the dashboard of other staff associated with the course.

(ii) Survey remains in “review” mode until all staff associated have reviewed it.

For the purposes of this project, it suffices to implement (i) as (ii) requires a trickier handling of the workflow.

### 11. Does this project need to follow MVC architecture?

The flask framework does not provide out-of-the-box support for a MVC based framework. For this group project, we expect students to show some degree of diligence in decoupling their components into Model, View and Controller components. We expect you show some degree of application of SOLID principles that you have learned in the lectures. Your lab and tutorial exercises should be a good guide on how to implement this. Think about the reusability and modularity of each component you build, Is this component capable of autonomous existence?

### 12. Can I create another survey for a course-offering for which a survey has already been closed?

Refer to answer 1. A course-offering can only have one survey.

### 13. Can staff add or remove questions from the pool of optional or mandatory questions?

Staff will not be able to add or remove questions from the pool of optional or mandatory questions. Only the admin can add questions to the pool and delete questions from that pool. When creating a new survey form, the admin can only add questions from the mandatory pool. The staff members do not have direct write access to the question pool and they can only add questions from the optional pool to survey forms in the Review stage.

### 14. Does every survey have the same set of mandatory questions?

The admin is responsible for creating a pool of mandatory and optional questions. When creating a new survey, the admin can choose **all** OR a **subset** of the mandatory questions to this survey. When adding a question to the survey, admin can choose what type of response the question might have. So, this means that each survey

can have varying number of mandatory questions and with different response types too. E.g a question such as “Do you like the timing of the lecture?”. This question could have a text-based response in one survey and a multiple choice in another

### 15. How is the notion of start and closing date of survey implemented?

As we have not introduced any multi-threading or event handling techniques in this course, the notion of time cannot be truly automated.

One approach that could be adopted for this project is to use python's time function get the current time and add "n" number of days at the point survey is created by admin. The survey continues to stay in "review" mode until reviewed by a staff member. When the application queries the database for available survey forms for a particular student or staff, it also checks the current time with the end dates of all the courses returned. If the end date is exceeded, the application changes the status of that survey form in the database and only shows the courses that are active.

For e.g., Let’s say, as a staff or student I log in and I am taken to my dashboard. The application will now query the database for courses that I am enrolled in. Let’s say it the query returns COMP1531, COMP2041 etc. The application then will check the end dates of these courses and update the status in the database if necessary. Only the active courses will be displayed on the dashboard.

So, in this scenario, there is a possibility that the survey goes into “close” state without having been reviewed or made open to the students. (This behaviour is quite similar to myExperience now, wherein there is a possibility that no respondents have filled out the survey form prior to the closing date). But for the purposes of this project, it can be assumed that a staff will review the course before the closing date.

### 16. There is no password or id for admin in the csv file. What are the credentials for the admin user?

As advised in the lectures, there will be no credentials provided in the **password.csv** for the admin user. You will be required to add these credentials and role=admin yourself to the database. The credentials for the admin are what you choose for the application (not necessarily, username=admin, password=password, but could be the same, but essentially the credentials you choose must be stored in the database)

### 17. Could you explain the format of passwords.csv? Why is there no admin credentials in passwords.csv?

This file has three columns: id, password, role. The “id” corresponds to the user-name, followed by the “password” for the user and then the “role” of the user, which is either staff or student.

e.g., 968, student527, student suggests that:

- user-name = 968,
- password = student527
- role=student

### 18. Will GitHub commits be part of the marking criteria?

For the final iteration, the tutors will be checking the github commits of all the project teams and we will be looking at each team member’s contribution. Currently, we see that there are some groups who have not have checked in any code to the repository or only one team member appears to have done most commits which signals unequal distribution of work. We would be checking github commits of all team members to ensure equal contribution to the project.

### 19. What package can I use for the metrics component of the survey (i.e. to visualise survey results)?

You could use the package: matplotlib. This is easily installed from python as:

```
pip3 install matplotlib
```

Link to website: https://matplotlib.org

### 20. What does the test-case “enrol student” mean?

This refers to the functionality of assigning a course offering to a student. You have been provided with a flat file `enrolments.csv` (a flat file) that shows a student_id, course,semester (e.g, 102, COMP1531,17S1).

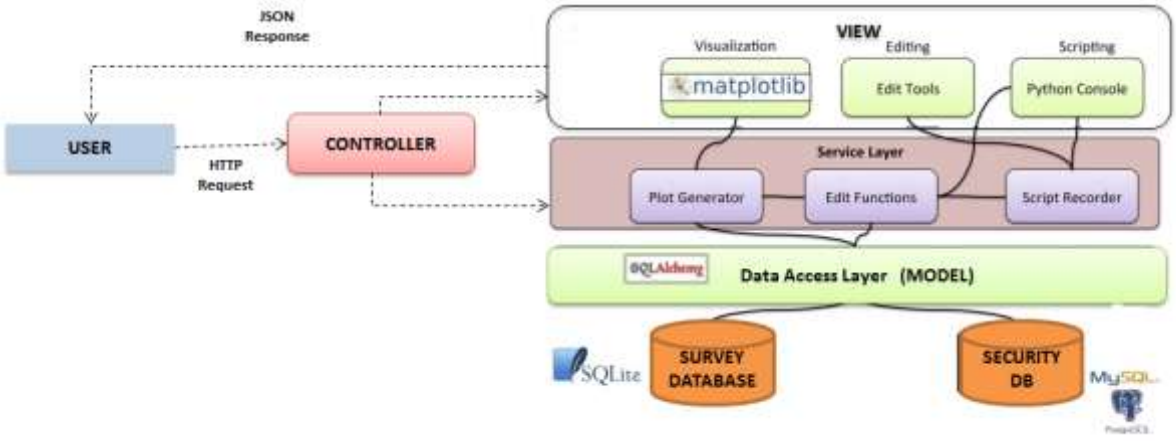
You would have a method that would load the data from `enrolments.csv` file into the database. We would like to see how this information is added to the database (and how well it is modelled in the database). Some students have modelled this as a single table with columns (student_id, course, semester). A better design would be to have two or more tables (think about, if you have 10000 students, 50 possible courses, 20 possible semesters, the former approach would result in duplication of information).

### 21. How do I submit the working software?

Submission details can be found in the file: give.pdf

### 22. Can you please give an example of a software architecture diagram

In the lectures, we looked at describing a high-level software architecture diagram using box-line technique, where the key components and protocols used are shown. Below is an example of a software architecture diagram for an arbitrary web application:




### 23. What is the difference between the test-cases that are needed to be submitted along with working software on Monday and user-acceptance tests due with final report?

As testing a complete a web application that includes both front-end and back-end testing is complex, we have broken the testing into two parts. The test-cases to be submitted along with working software on Monday, will consist of testing back-end functionality similar to the sample test-cases provided to you (e.g, successfully assigning a course to a student etc). The user-acceptance tests will help us to validate your front-end functionality (e.g, what happens if a respondent tries to submit a survey without filling all the responses. The expected behaviour should be defined in your acceptance criteria). So, we will be logging into your application (based on credentials provided in `enrolments.csv`) and testing the various user-stories and associated acceptance criteria defined by you.
In particular, automated front-testing for FLASK web app is not needed.