Rules to follow:

1. Individual project, for which you must submit :
    a. A .zip file containing the code and any other file necessary (ansible playbook, bash script, …)
    b. A .pdf report – not a powerpoint presentation – to explain your project goal, the technologies you chose, why they are a good fit for your project, …
    c. A pre-recorded video of your website (with « obs studio » for example)
2. Deliverables MUST be in English (code itself, code comments, pdf report, …)
    a. Course is officially in English, so I can only evaluate work in English
    b. Must be submitted on August 22nd at 23:59 the latest to g3d@ecam.be
    c. If the files are too big, you can send them through « wetransfer » or sharepoint to g3d@ecam.be
3. Special accomodation ? Contact me in advance to see if I allow it. Any justification after the deadline (or very close to it) will be considered as invalid.

While the project guidelines are roughly similar to the 1st session, the grading will take into account this is an individual project. There will be no oral presentation.

Basic features (+ perfect pdf report -> 14/20) :

1. Create a web application of your choosing (e-commerce, accounting handling, calendar management, …)
    a. Each project must be different from each other
    b. Having 1 000 lines of (python) code for the back-end (excluding any front-end development) of the "basic features" could give a rough estimate of the complexity of the application
        i. Lines of code will NOT be a metric for scoring
2. Backend
    a. Does not need to be a Restful API (only implement it as API if you plan to use it)
    b. Must be developed in Python, any framework is allowed (flask, django, …)
3. Basic UI
    a. Can be developed in any language
    b. Pretty front-end will not be part of the score, but it needs to be usable
4. Scalable application
    a. Can your application scale for 100m+ users in your database, 1k+ concurrent users while still being fast? What about the elements (objects, posts, images, …) created by your users?
    b. How would you scale (only for the database)? How could your code handle concurrent writes (does not need to be implemented)?
5. Must use at least one nosql database
    a. Its use must be justified (why use it vs a sql database? Or vs another nosql database presented during the course)
    b. Feel free to use any nosql database, even if not seen within the course.
        i. Keep in mind you are creating an application for a big company, so you should be careful about Licensing (free to use?), maintainability (is the

database still under development by more than a few guys / is it financially supported by some big company / …), how easy it is to find developers…

Expected content of your pdf report:

1. Explanation of the software architecture (interaction with the database) – 10-15 minutes (including Questions & Answers) or more if necessary
    a. Why did you choose this database?
        i. How do you use its features?
    b. Other (relevant) databases you considered and dropped (why)?
    c. How do you scale for more users?

Extra scope (if done should be quickly shown in the video):

! Only considered if the basic scope is fully implemented

! You do not have to implement everything; this is merely a list of ideas. Implementing only a few features "perfectly" with a very good pdf report might give you 18/20+ depending on the complexity.

1. Restful API with front-end integration
2. Dockerfile & docker-compose to build & run quickly your code
3. Choosing a database not seen within the course, but being able to justify why it is a better fit
    a. You should be able to easily explain how this database is different, why is it better, …
    b. Some ideas of databases (feel free to find others): https://nosql.mypopescu.com/kb/nosql
4. How do you deploy your database for scaling?
    a. Technology: Ansible, python scripts, bash, …
    b. Server configuration?
    c. Sharding?
5. Could you switch your database with another one easily?
    a. Interface to interact with the database, and showing a switch from a database to another
6. How do you handle data migration (rename of a field, change of a table name, …) when you have terabytes of data and your application must remain up?
    a. Example of data migration script, how is it launched, and showing in the code how you handle the migration occurring real time while you still are displaying data to a user
7. How do you handle rollback during a failed data migration?
8. If the database allows it, handle a read-replica as much as possible in your app to reduce the load. How do you handle stale data (if it is important in your code)?
9. How do you handle schema validation? What happens if the database contains incorrect data that your application is not able to read?
10. Do you handle transactions? How did you implement them (even if the database does not support it you might want to implement on the application level)
11. … (You are free to find other interesting features to implement)