Coursework in Cloud Computing

Informations and Details

Summer Term 2025

Please make sure, you have read and understood this text!

Organization

In the moodle course, you find an Excel sheet where each participant is assigned to a combination out of Application, Image, Programming Language and Database Platform. This assignment is done by Random using a Python script. Be aware, that all combinations have the same level of difficulty.

Use the fact, that there is always a number of students who has to use the same Image, the same Programming Language or the same Database Platform and there is also always a number of students who have to provide the same Application. Just the combination out of all four aspects is indivudual. Use this fact to find out things together in groups and communicate with different people about different aspects of your Coursework.

Example: To find out who has to use the same Database Platform as you, simply use the filter option within the provided excel sheet. Find together as a group and try to figure out how to use it for your projects.

Please go to https://gitlab.hof-university.de and create an account or login with your already existing account. Create a project called "cloud_computing_2025". Create it as an private project, but dont forget to add the User "wwiedermann" as a Member of your project. After that, paste the URL to your project to the relevant database control within the moodle course.

In the end, everybody has to create an own solution, because the combination of technologies is unique per student. Try to make your own effords visible by using frequent pushes to gitlab.

That means, a good solution with frequent commits pushes will get better grades than a good solution with very less commits and pushes, because this reveals differences in the quality of the traceability of the own work and the competence in dealing with the development tools used.

Applications

In this section you get additional information about the applications you have to build

The frontends of all three Applications will be based on two or three webpages. There is no need to take much efford in the design of those pages. Keep it simple. The core focus is on the virtual infrastructure you have to build up for this App.

You can keep the frontend of the app simple, that means, there is no need for client side javascript at all. You can do everything by server side rendered html if you want. (But there is also no minus if you do it more flexible, but it wont help for better grades.)

Appointment coordination

Think about Doodle, the DFN-Terminplaner and so on, but just in a very limited and minimized way. The App has three pages. One page to create a new appointment coordination, one to reply to an appointment coordination and one to see the results of an appointment coordination.

To make it even more simple, an appointment coordination allows to offer between one and three appointment suggestions. So the controls for those three suggestions can be placed statically within the page to create a new appointment coordination. There is no need to allow to dynamically add additional suggestions.

QR code based survey app

Think about the idea of offering an QR code to the participants of your lessons, where they can do a live voting during the lessons by scanning the code and doing the voting using their mobile phones.

To keep it simple, the Application should have one page to allow a lecturer to create survey. The type of survey is limited to ask for a question to be answered with yes or no. As an result of those create survey action, the lecturer should get as well an QR code as an exportable graphic and a link to view the results of the survey later.

The results of the survey should be visualized als a diagramm with two bars – one for yes and one for no. Dont forget to show the question answered near to the diagram so everybody can understand what the participants have been asked.

The third page is the page an attendee gets if he/she scans the QR code. It should show the question two radio buttons to select yes or no and a button to submit the vote.

Public notes platform

Think about pintrest for textual notes. So the system should offer one page to create individual notes which should be marked with individual tags. Another page should offer a search form, where users can search for notes (by using the tags) and show the results of the search. The third page should show notes of on distinct user. A user should be identified by its email addresse.

There is no need to offer real user authentication, a textfield where to place the own email address while creating notes is just enough.

Image

Next column after the application you got assigned by random is Image. That means that all the containers you will build to deploy your application have to be based on that image. Your webserver container, your database container and so on. So even if there is an official image for for example postgresql, you are not allowed to use it. You have to setup your own postgresql container based on the image from the image column.

If there is debian/ubuntu in the image column, you can use either debian or ubuntu as an image. There is no debian/ubuntu image avaliable. Thats different to opensuse/leap where you cant choose between opensuse and leap because leap is the stable distribution of opensuse (opposed to opensuse/tumbleweed as the more bleeding edge one).

Programming Language

The next column ist about the programming language to use. To make it simple, i suggest the usage of the framework Flask when using Python, to use simple apache mod_php when being applied to php (of course you can also use Laravelle if you already know how it works). With javascript use nodejs with express.js (if you are not already used to another framework). With Java you could choose plain JSP or SpringBoot as you prefer.

If you already have knowledge with a webframework for the programming language you got assigned, choose that framework. Otherwise try the ones i suggested.

That means you have to do write your webapp with that language, but you also have to setup an environment where this app later can be deployed to. The setup of the environment is the more interresting part in respect of cloud computing.

Database Platform

Setup the database platform you got assigned to and do this based on the image you got assigned to.

The database has to be used to store the data from your application. It should be setup in a way that redeployment of the database container will not cause data loss.

So what to do?

You have to setup a virtual deployment infrastructure based on Docker and Docker Compose to run your Application. It consists out of one loadbalancer node based on haproxy (also to be set up based on your assigned image), two webserver nodes where your web application runs and one database node where the data is stored. The connection between webserver and database should be done on an isolated network (where the loadbalancer is not in).

In the end the web port of the loadbalancer should be bound to all interfaces of your container host (usually your laptop) and the database port of your database should be bound to your localhost interface to be able to directly connect it from your laptop using a database client (like for example pgAdmin4 for postgresql or mongosh for MongoDB).

After it works fine for Docker Compose, setup the same infrastructure (you can simply reuse the self generated images) using kubernetes using minikube, k3s or kind.