## 2 work order

HerPractical work should have the following resultsdeliver:

- oneexecutableApplication with Flask and aDatabase
- oneDocumentation for the application
- onePlatform on which the application is deployed in the cloudbecomes
- oneDocumentation for the platform

## 2.1 Application

The most important delivery item is aexecutable web application. You put them in a by the deadlinefor the examineravailable Cloud environment ready and there it will be transferred from the Examiner to thefulfillment therequirements tested.For the accessabove the web browser and via API with http or httpswill a URL with IP address (or DNS name), the portfor the access as well as therequired Log in-information(fortheFlask application) documented. The application must be submitted by the deadlinefor at least 4 weekswhiletheCorrection time fordisposal stand.

### 2.1.1 Type and functionality of the web application

HerWeb application provides the user with afunctionalityto theDisposal,the youinsideone determine yourself within the given frameworkbe able.She has to be therethe following requirementsfulfill:

- there is an interactive oneweb interface,with the theuser input,processing or Actionstriggerandcan view results.
- theAccess to the web application is requiredabove user accounts withpasswords regulated will. new userhave towitha unique username andEmail Account to register.
- theIn addition to the user accounts, the application must store data thatfortheapplication are specific (example: recipes, comments, employees, projects,etc…).shewillin one relational database stored.
- there is one of its ownbusiness logic. This can include calculations, actions,reviewsfrom inputs or just the collection and display of data.
- onselected Web application data must be in addition to the interactiveweb interface also (at least) read accessabovea RESTful web APIpossible be. That's truefor a specific content of the respective web application (i.e. not just user data)

### 2.1.2 Technologies

The following technologies can be used:

- Database system: MySQL, MariaDB orPostgreSQL
- Programming language: Python version from 3.6
- Web framework: Flask, various Flask extensions
- Web server: Gunicorn (standalone) or in combination with Nginx or Apache

### 2.1.3 Source code

Of theSource code of the application will beCompletelyasRepository on GitHub, Bitbucket orGitLab provided. It is randomly selected by the examinerchecked.The URL andat most users/ passwordfor read access must be documented.

### 2.1.4 Tests

Tests with which thefulfillmentthemain functional andquality requirements checked willbe able, have to be documented. Thisbe able descriptionsfor manuallyexecuted be tests or automated tests.Forthetests, a short test protocol must be drawn up,the documents what requirements have been met so farFulfillswere and which were not. It doesn't matter thatEveryone tests are successful. Rather, the aim is to show that the tests are defined andcarried out became.

## 2.2 Infrastructure

### 2.2.1 Technologies

The following technologies can be used:

- Fortheworkallowed to allx-aaS services are used (e.g. IaaS, PaaS, SaaS, DB on-premisesEtc.). The technologies should make sensechosenthat will add valueview thedevelopment and/or operational operation of the application.
- Container Registry: If you create your own container images,have tothiseither publiclyatDocker Hub or apossibility are offered tothis to use images (key / account with reading rights on the corresponding Dockerhub repository)

### 2.2.2 Infrastructure as code

theInfrastructure is supposed to be through scripts and configuration files as far aspossible automated be rebuiltbe able. This will be randomly selected by the examinerchecked.

## 3 Structure of the practical work

Of thewritten partcontainsthe followingSections:

### 3.1 Management summary

- Summary describing the project, its objectives and the results achieved

### 3.2 Application

- Brief description offunctionalityandOperation of the web application ('UserManual')
- List of goals and the most important onesrequirements
- Description of the architecture (diagrams, each withexplanatory Text), in detail:
    - Web application data model as ERD or class diagram, with shortexplanatory text
    - processesin the web application asactivity, State or sequence diagram withshort description
    - Provision of the components (e.g. application, web server, DB server,additional tools) as Diagram with description.•

test log

- Short documentation of the API interface

### 3.3 Infrastructure

The following points are required in the documentation

- A short onesolution descriptionthe infrastructure(s) / cloud usedSolutions) / x-aaS and a shortReason fortheChoice.
- OneArchitectural drawing of the infrastructure with theTherefore required core components(the Infrastructure / Cloudsolution / x-aaS) thefor the operation of the applicationselected became.
- Of theadded value intended or gained through the selection of the infrastructure / componentsgot to be clearly documented.
- Iffuturechallenges are expected orwhiletheimplementation occurredare

this short describe and document how to deal with itcould.

- considerationsfor scalinghigh availabilityandporting(Possibility on aother infrastructure to migrate)have to to be detained.
- Script and source code:All Scripts (and if necessary data) for replicationthe vicinityneeded will,have to Completely Deployed on GitHub, Bitbucket or GitLabbe (same place where the app source codeavailableis). A link to this must be in thedocumentation be deposited.

- Outof Scope - The following points are explicitly NOT expected andbelongNOT in the infrastructure documentation
    - Installation instructions (e.g. installation of Docker, Docker-Compose, Kubernetes, OS, database, web server, other software, etc.)
    - Stepfor Step-by-step instructions: e.g. how is component y obtained in the cloud x / created / deployed / configured etc.
    - Basic descriptions of the technology (e.g. what is virtualization, what is object Storage)
    - script/ code examples thefortheCreation of the infrastructureneededwill:Belong Not in thedocumentation

theInfrastructure documentation includes a maximum of 5(five)A4 pages.

### 3.4 Appendix

- User documentation web application (instructions for use)

Of theThe length of the document, excluding management summary and appendix, should be at least 8 and may not exceed 830
pages amount.FortheFor the correct structure of the document, refer to the guidelinesfor written
Working in ipso education.

## 4 Submission of your practical work

Upload your practical work on time as a«solution document»(in pdf format),all scripts andthe associatedSource code in one«ZIP File»as"Appendix" onto the school platform.
To thesubmissionheard as well as providing the corresponding files on GitHub, Bitbucket or GitLab.