EPITA0121
April – May 2021

Group Assessment (CA)

Module Title: **Back-end Web Development**
Module Code: EPITA0121
Assessment Type: Practical Assessment
Weighting: **100%**
Maximal Possible Mark: 100 marks
Final Submission Date: 25/05/2021
Weekly Submission Dates start: 09/04/2021 or as soon as your team is ready.

Assignment brief

Assignment title	***Concepts, design and implementation of a REST API in Cloud***

Purpose of this assignment
To enable learners to understand the concepts of complex REST API and apply the skills acquired to develop and test the project.

Groups must commit weekly updates to GitHub starting Friday, April 9th. The commit message (or build notes) must note what has been achieved that week and what is planned for the following week.

Groups must use Trello to track their 'sprints' each week. The Group leader will keep the Trello board updated and present it each week in class. The lecturer will explain how to use Trello. It is up to the Group leader to decide how much they want to commit and deliver each week to reach their deadline.

Alice and Bob are business partner. They came up with a business idea of building a new platform for job seekers. They want to do something different to similar services that already exist in the market (i.e., LinkedIn, Indeed etc).

As a group of developers led by a team lead, the task for your team will be designing and developing the new system and deploying to any public cloud provider. Your team is responsible for building only the backend of the system which should be ready to be plugged in to any frontend client.

As per the business requirements, your Application should be able to carry out following tasks:

API Specification:

- It will be a REST API with GET, PUT, POST & DELETE methods
- API must have public(open) and private(secured) endpoints.
- System should be able to register users (job seekers & employers) where they can login to the system. Users should be able to reset their password if they forget the password (confirmation via email) or if they just want to change the password.

- Ability to update user profile, skillset, availability for a job etc.
- Ability to advertise themselves in the system to potential employers.
- Employers should see all the candidates and be able to connect to a job seeker, see his/her profile, skillset and communicate with them via messaging system (not Chat, but just exchanging message with potential candidates in the system).
- Candidates shall have the ability to accept or reject an offer from employees.
- Candidates can also visit an employer profile, see their jobs and choose to apply for a job.
- Employers can pay fee to bump up their advertisement in the platform (showing their result on the top of the search). You are not expected to implement the payment system, but you should have an option to classify advertisements as premium version and free version.
- After the job is done, employer can pay employers in the system, rate their work and finally mark the job as done; candidates should be able to rate an employee as well.

There should be Admin (Super user) functionalities in the system as specified below:

- Admin can ban a candidate and employer.
- Admin can soft delete or hard delete a candidate/employer.
- Admin can update any candidate's or employer's profile as well as their advertisements.
- Admin can remove a job advertisement from the platform.

Tasks

1. You will be using C# .Net Core Framework to develop a REST API
2. Complete and deliver a UML diagram showing the objects, models and actors in your design.
3. Swagger API documentation is a must have, so that the API can be tested in local machine using Swagger.
4. You can choose any Database (MySQL or SQL Server) for persisting the data in the System/Server.
5. Complete and deliver a working version of your software
6. Complete and deliver any configuration files (appsettings.json)
7. All projects should be submitted to GitHub and show multiple commits and branches from the team over the duration of development.
8. A short report (About 2500 – 3000 words) explaining your project, describing the areas you have completed. must highlight all areas you have tackled in your project (see API Specification) and provide pointers to the code where you have demonstrated the various features.
9. The API must be deployed in a public cloud platform and live URL should be supplied with project submission; your GitHub repository readme should contain the link to deployed API.

Follow up questions / clarifications

You will need to make many assumptions as you develop the project. You should document these assumptions in your report, so they are considered during assignment evaluation.

It is each team's responsibility to get in touch the course lecturer if you or your team require any clarification on the specifications outlined above.

Grading Marks:

1. Codes: 75%
2. Report: 25%

Marking Criteria:

1. Project structure, coding style, using of well-defined functions and methods in the controllers/API endpoints.
2. Code comments.
3. Weekly GitHub pushes
4. Deployment to Cloud
5. Clear and well documented report with a brief explanation of the project specification, designs and its features.