COM673 Sample Assessment

Read all the instructions below and all the questions before attempting the solution

Instructions

- This is a 3 hour, open book assessment
    - You may use any non-human resources, online or offline, to help you
- You may **not** communicate (in person or digitally) with any other student or person, except the invigilator
- Do not write your name anywhere in the source code you submit
    - Use your student number to identify yourself e.g. B01234567
    - Put our student number in a comment in the first line of your code and also in the readme file.
- There are no penalties for asking a question, please clarify any questions you have
- Read the entire question before beginning

Problem Description

KNA Bank have asked you to develop a web application to manage their employee database. Employees are organised into the following departments:

- Investments
- Loans
- Human Resources
- IT Services
- IT Support

The database includes the following details for each employee:

- Employee ID
- Department (ID)
- Name
- Address
- Email
- Telephone
- Starting year
- Qualification
- Experience (Description)

An employee can work for one department at any time but may be moved between departments.

The Application

The application should include the following features:

1. The application should present managers with a list of all employees, organised by department.
    a. Show a list of departments which can be used to filter or only display employees for that department
2. Show a summary list of employees in a table (one row for each)
    a. Clicking an employee name should open another page showing full details with an option to update or delete that employee.
3. Include a link/ button which will allow a new employee to be added.
4. Allow searching of employee for a keyword in name, qualification, and experience description.
5. Format and layout the web app pages using BootStrap CSS.

Azure Setup

1. Create a free tier Azure web app in your student account configured for Java and Tomcat
   a. Use the following name format for the Azure web app b01234567-exam2019, replacing the red example id with your real student id.
   b. Create a new App Service Plan for your web app with a name based on your id, e.g. B01234567-exam2019 in North Europe on the Free pricing tier
2. Create an Azure SLQ database, again named using our id, e.g. B01234567-exam2019.
   a. Setup the database with the required tables and add some sample data.

Submission Guide

If you can't get something working, submit / upload your attempt anyway. Don't spend too long on any one issue, move on to something else and come back to it later.

1. Submission is on GitHub.
2. Push to master from terminal in VS Code (or upload in github.com). If uploading in github.com ensure that you **Commit to Master Branch** (at the bottom of the GitHub window when uploading.
3. At assessment due date, the files uploaded are locked for the submission (don't edit the files after that date, as it will email me each time and we can only grade the Commit at time of assessment end)

Upload & Commit:

1. The Spring Boot Java project
2. The SQL schema you used to create and populate the database (just save the Create Table and Insert SQL statements to a text file called Schemal.sql). Note: It is OK to use a web tool to generate the SQL if you wish.

Marking Scheme

1. Create a new Azure Web app called your student number-exam e.g. "B01234567-exam.azurewebsites.net". Create a new App Service Plan for your web app with the B01234567-exam in North Europe on the Free pricing tier **(5 marks)**
2. Configure the app for Java and Tomcat **(5 marks)**
3. Create a SQL database called **B01234567-exam** on the Basic pricing tier in the same resource group **(5 marks)**
4. Populate the database table based on the data above with rows of sample data (as given above) **(10 marks)**
5. Create a Spring Boot app with the following features:
   1. Uses the groupID: **ie.ulster.exam**
   2. Add a readme file in the root with your student/ exam id and any instructions.
   3. Connects to and loads the data from the Azure database **(5 marks)**
   4. Converts the data from a resultset to a List of objects **(5 marks)**
   5. Creates the UI items listed above (marks for adding them and marks for getting each aspect asked in the question implemented) **(15 marks)**
   6. Add, edit, and delete functionality **(15 marks)**
   7. Search function **(10 marks)**
   8. Neatness, bracket formatting, code quality, meaningful comments and readability **(10 marks)**
6. Deploy the app on Azure (you can use FTP or any other way to upload the .war file) **(10 marks)**