

ĐỀ THI THỰC HÀNH MODULE INFRA (Thời gian làm bài: Tối đa 01 tuần)

1. Bài toán:

Cần xây dựng hệ thống quản lý nhân viên cho công ty IT.

Giai đoạn 1 chúng ta cần quản lý thông tin của nhân viên và quản lý thông tin nhóm.

Biết rằng

Thông tin **nhóm** sẽ bao gồm tối thiểu:

- + Tên nhóm
- + Ảnh đại diện
- + Giới thiệu chung về nhóm

Thông tin **nhân viên** sẽ bao gồm tối thiểu:

- + Tên nhân viên
- + Email: duy nhất
- + Ngày tháng năm sinh
- + Giới tính
- + Ảnh đại diện
- + Ngày vào công ty
- + Ngôn ngữ lập trình: 0 hoặc nhiều
- + Giới thiệu nhanh (do bộ phận nhân sự tự nhập)
- + Bản thân tự giới thiệu
- + Nhóm: 1 hoặc nhiều

2. Yêu cầu 1 (frontend)

Thiết kế màn hình quản trị thông tin cho nhóm, nhân viên.

Sử dụng Bootstrap và Angular.

Cần có bản thiết kế thô (vẽ bằng tay hoặc sử dụng Balsamiq) hoặc **HTML mockup**.

Các trang cho quản trị viên:

- + Đăng nhập: thông tin đăng nhập lấy trong DB
- + Quản trị nhóm: CRUD
- + Quản trị nhân viên: CRUD, cài đặt lại mật khẩu, gửi email thông báo mật khẩu cho nhân viên
- + Confirm trước khi gửi yêu cầu xóa

Các trang cho nhân viên:

- + Đăng nhập: thông tin đăng nhập dùng email nhân viên và mật khẩu do quản trị đặt
- + Xem thông tin cá nhân của bản thân
- + Sửa thông tin: cho phép sửa ảnh đại diện, tự giới thiệu bản thân
- + Đổi mật khẩu

Yêu cầu	Nội dung	Điểm
1	Đăng nhập	5
2	Quản trị nhóm	30
3	Quản trị nhân viên	40
4	Gửi email	5
5	Xem thông tin cá nhân	5
6	Đổi mật khẩu	5
7	Sửa thông tin	10

3. Yêu cầu 2 (backend)

- + Thiết kế DB cho hệ thống: yêu cầu có ERD và tài liệu giải thích chi tiết từng bảng và trường trong bảng
- + Lập trình thực hiện các API đối ứng cho các tính năng của frontend ở trên
 - + Dùng MySQL cho cơ sở dữ liệu
 - + Có thiết kế API cho các API, sử dụng JSON:
 - + Input params
 - + Output
 - + Error
 - + Đăng nhập sử dụng oauth hoặc tự phát hành token có thời hạn cài đặt được trong config
 - + Dùng SMTP của Gmail để gửi email
 - + Bật **Less secure** khi test bằng email của cá nhân tại **@nal.vn**
 - + Source code không để thông tin tài khoản
 - + **Chú ý:** tính năng xóa thực hiện **xóa logic**

Yêu cầu	Nội dung	Điểm
1	Thiết kế DB	15
2	API 1 (đăng nhập)	10
3	API 2	20
4	API 3	25
5	API 4	10
6	API 5	5
7	API 6	5
8	API 7	5
9	API khác	5

4. Yêu cầu 3 (infra)

- + Vẽ sơ đồ thiết kế hệ thống và tương tác giữa các thành phần trong hệ thống
- + Viết các file Dockerfile để tạo docker image cho frontend, backend

DEPLOY: thực hiện 1 trong 2 tùy chọn hoặc cả 2.

- + **Tùy chọn 1:** Viết file docker-compose.yml cho toàn bộ hệ thống theo sơ đồ thiết kế, loại trừ các thành phần sử dụng bên ngoài như SMTP. Yêu cầu sử dụng:
 - + `http://IP/` sẽ truy cập vào frontend
 - + `http://IP/api` sẽ truy cập vào backend

+ **Tuỳ chọn 2: Kubernetes:**

- + Đưa các docker image của frontend/backend lên GCR
- + Viết các file cấu hình cho deployment các thành phần hệ thống trong thiết kế trừ các thành phần sử dụng bên ngoài như SMTP
 - + Các thông tin cấu hình, API truy cập sẽ cài đặt ở biến môi trường
 - + thông tin kết nối sử dụng secret
- + Viết các file cấu hình cho service (loại ClusterIP) frontend, backend và các service khác nếu cần
- + Viết các file cấu hình cho secret như nội dung username/password để kết nối tới SMTP, DB. **Chú ý không commit thông tin thật của mình.**
- + Viết các file cấu hình cho PV/PVC sử dụng bởi hệ thống
- + Viết duy nhất 01 service thuộc loại LoadBalancer, mở cổng 80 để truy cập từ ngoài vào hệ thống.
 - + http://IP/ sẽ truy cập vào service của frontend
 - + http://IP/api sẽ truy cập vào service của backend

Yêu cầu	Nội dung	Điểm
1	Sơ đồ hệ thống	30
2	Dockerfile cho frontend	10
3	Dockerfile cho backend	10
4	docker-compose.yml	100
5	K8S deployment	30
6	K8S PV/PVC	10
7	K8S service ClusterIP	10
8	K8S secret	20
9	K8S service LoadBalancer	30