**Mô tả kết quả đạt được của bài test**

Bài tập 1: Viết frontend nền tảng web hiển thị 1000 hình tròn với đường kính từ 10 tới 100 pixel chuyển động với quỹ đạo bất kỳ, vận tốc > 10pixel / s. Hiển thị FPS, yêu cầu >= 30. Có thể thử dụng thư viện.

Link github: <https://github.com/sang18112001/VHT_Ex1>

Mô tả: Bài tập đã thực hiện được việc hiển thị 1000 hình tròn theo yêu cầu đề bài đặt ra. Quá trình thực hiện như sau:

* Khởi tạo: Mỗi hình tròn được tạo dựng qua việc sử dụng css cho mỗi một thẻ div (được tạo với javascript) với width random theo yêu cầu (chính là đường kính của đường tròn).
* Di chuyển: Công việc này được thực hiện bằng việc đưa vào góc muốn chạy (hướng chạy) và tính được giá trị cần cộng vào mỗi frame theo 2 trục tọa độ.
* FPS: FPS được tính bằng cách đếm số frame đã render trong mỗi giây. performance.now() sử dụng để đo thời gian giữa các frame. Khi đã đủ một giây, số frame được chia cho thời gian để đạt FPS. Kết quả được hiển thị trên giao diện người dùng.

Bài tập 2: Viết một hệ thống gồm 1 frontend nền tảng web và 1 backend ngôn ngữ tùy chọn thực hiện các nhiệm vụ sau:

* Hiển thị trên frontend khung đăng nhập thông tin gồm có: username, pasword
* Frontend kết nối tới backend để xác nhận username và password, thông báo thành công / không thành công
* Notes:

1. Frontend không gửi password dạng clear text tới backend (cần mã hóa hoặc phương pháp khác)
2. Frontend kết nối tới backend sử dụng kết nối bảo mật
3. Cơ sở dữ liệu lưu thông tin người dùng sử dụng sqlite. Không lưu mật khẩu dạng clear text trong CSDL.

Link github: <https://github.com/sang18112001/VHT_Ex2>

Mô tả: Bài tập được thực hiện với chức năng đăng kí, đăng nhập và sửa đổi thông tin tài khoản với yêu cầu không gửi clear-text password lên từ frontend sang backend và không lưu chúng trên database.

1. Frontend

* Công nghệ sử dụng: ReactJs.
* Cho phép người dung thực hiện đăng kí và đăng nhập qua form.
* Dữ liệu về mật khẩu đã được mã hóa và gửi sang phía của backend.
* Sau khi người dung đăng nhạp, họ có quyền để sửa đổi thông tin của chính họ.
* Tuy nhiên đối với những tài khoản mà không có quyền admin thì không thể truy cập vào trang dashboard để xóa đi thông tin người dùng.

Tài khoản admin đã được tạo để sử dụng:

Username: *admin*

Password: *1213456*

* Sử dụng redux-persit để hỗ trợ lưu thông tin không quan trọng của người dùng trong local storage.
* Sau khi người dùng logout sẽ xóa đi thông tin đó trong local storage.

1. Backend

* Công nghệ sử dụng: NodeJs, SQLite
* Backend đã thực hiện được chức năng authentication và authorization.
* Tạo được chức năng đăng kí cho người dùng qua việc lấy mật khẩu đã được mã hóa từ phía frontend và lưu chúng trong SQLite.
* Qua việc sử dụng jwt, khi người dùng đăng nhập thành công, thì sẽ nhận được một mã accessToken (được lưu trữ ở trong localstorage) và một mã refreshToken được lưu ở trong Cookie.
* Mã accessToken được sử dụng cho việc người dùng đăng nhập. Khi người dùng đăng nhập, password sẽ được gửi sang backend để so sánh với password trong database. Nếu chính xác, server sẽ trả về một accessToken. Với mã này, người dùng có thể thực hiện xem hoặc sửa đổi thông tin tài khoản hoặc xóa tài khoản(chỉ với admin).
* Mã refreshToken được lấy ra từ cookies để tạo ra một mã accessToken mới cho người dùng, khi accessToken bị hết hạn (mã này phải để thời gian hết hạn để bảo mật thông tin người dùng). Khi mã refreshToken cũng hết hạn sẽ cho phép tài khoản Logout.