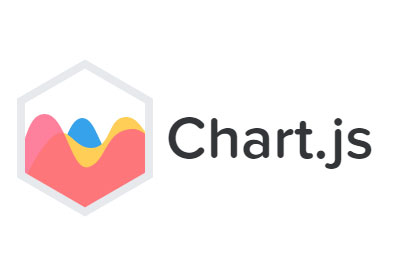
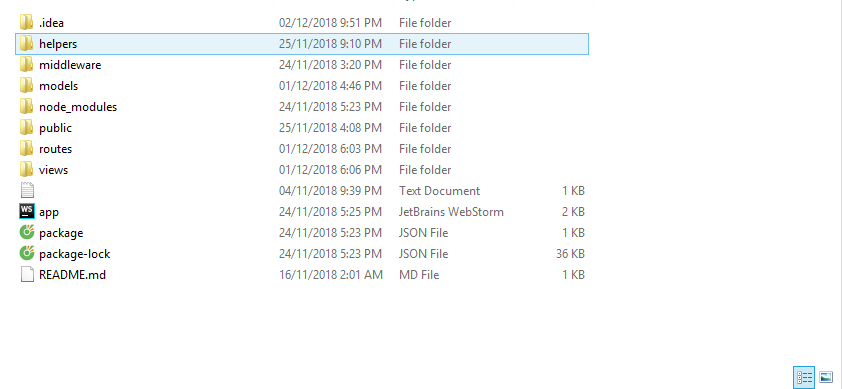
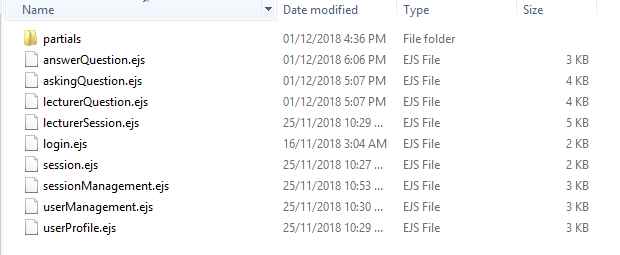
**BÁO CÁO BÀI TẬP LỚN**

Nhóm: 8

Đề tài: Q&A

1. Công nghệ sử dụng
   * Front-end
     + Sử dụng HTML, CSS, JS thuần
     + Thư viện w3.CSS để thiết kế trang web  
       
     + Thư viện fontawesome để sử dụng các biểu tượng (icon)  
       
     + Thư viện axios để tạo AJAX request
     + Thư viện chart.JS để tạo biểu đồ trong phần thống kê khảo sát  
       
   * Back-end
     + Sử dụng Node.js làm lõi  
       
     + Sử dụng framework express.js để viết các route (đường dẫn)  
       
     + Sử dụng socket.io để tạo ứng dụng thời gian thực  
       
     + Sử dụng thư viện mongoose để kết nối với cơ sở dữ liệu mongoDB  
       
   * Phần cơ sở dữ liệu
     + Sử dụng MongoDB làm cơ sở dữ liệu  
       
2. Cấu trúc project



* Trong đó:
  + File App.js là nơi chứa những phần chính được lắp ghép từ các phần của ứng dụng và là file chính để chạy ứng dụng
  + Thư mục views:  
    
    - Là thư mục chứa những tệp ejs (cũng tương tự html, thêm việc hỗ trợ việc hiển thị dữ liệu từ server) – là 1 view engine
  + Thư mục routes:
    - Là nơi chứa các file js được viết để làm đường dẫn khi truy cập vào ứng dụng
    - Có 5 file js:
      * Index.js là file chứa các route chung của ứng dụng (đăng nhập, đăng xuất, …)
      * Các file khác chứa các route cho các api của ứng dụng, khi thực hiện request tới server để lấy dữ liệu thì sẽ gọi tới các route được định nghĩa trong các file này.
  + Thư mục public:
    - Là nơi chứa các tài nguyên của ứng dụng (các file css, js) giúp các file ejs hoạt động được như một trang động và được styling
  + Thư mục models:
    - Là nơi chứa các Schema, Model của cơ sở dữ liệu
    - Ứng dụng gồm 4 loại Schema, Model:
      * Comment
      * Question
      * Session
      * User
    - File index.js là file cấu hình mongoose và xuất các model bên trên để sử dụng trong thư mục helper
  + Thư mục middleware:
    - Chưa file index.js định nghĩa phần trung gian để có thể truy cập vào một route nhất định – đóng vai trò như điều kiện để có thể truy cập vào 1 route (VD: người dùng chưa đăng nhập thì không thể truy cập vào trang có sessions, giảng viên không thể truy cập vào trang của admin…)
  + Thư mục helper:
    - Tương tự như phần controller trong mô hình model-view-controller
    - Chứa các phần tương tác với cơ sơ dữ liệu, thực hiện các phần create, read , update, delete bản ghi trong cơ sở dữ liệu

1. Phân công công việc:
   * Nguyễn Mạnh Dũng:
     + Viết back-end, tạo cấu trúc project, Tạo ứng dụng hoạt động theo thời gian thực (các trang tạo câu hỏi, tạo câu trả lời, quản lí các phiên hỏi đáp + tạo phiên hỏi đáp…)
     + Chỉnh sửa html, css, js cho toàn bộ các trang
     + Tạo model Question, chỉnh sửa các model khác
     + Kết nối các thành phần của ứng dụng với nhau…
   * Nguyễn Khắc Vũ Hiệp:
     + Viết phần html, css, js cho trang tạo câu trả lời, quản lí các phiên hỏi đáp
     + Tạo request tới server sử dụng axios
   * Phạm Minh Huyền:
     + Hỗ trợ viết phần js cho trang tạo câu trả lời, viết phần html, css, js cho trang tạo phiên hỏi đáp, hiển thị toàn bộ phiên hỏi đáp
     + Tạo model Session
     + Tạo request tới server sử dụng axios
   * Trần Thị Quỳnh Giao
     + Viết html, css, js cho trang tạo câu hỏi, trang đăng nhập, trang chỉnh sửa thông tin cá nhân, tham gia chỉnh sửa js trang quản lí người dùng
     + Tạo model User
     + Tạo request tới server sử dụng axios
   * Tống Lý Trinh
     + Viết html, css, js cho trang quản lí người dùng, chỉnh sửa js, tương tác thời gian thực phần thêm câu trả lời
     + Tạo model Answer
     + Tạo request tới server sử dụng axios