**BE - NESTJS**

1. **Kiến thức về ngôn ngữ lập trình - JavaScript/ Typescript**

* Cài đặt môi trường
* Các khái niệm cơ bản: biến, kiểu dữ liệu, vòng lặp, toán tử, điều kiện
* Các khái niệm nâng cao: Hàm, callback, promises, async/await, object, class, string method, number method, array method, scope, json v.v
* Typescript: class, interface, enums,  type alias, generics

1. **Khả năng sử dụng công cụ**

* Git, gitLab
* Postman
* IDE, DBMS
* Package manager: Npm, yarn
* Eslint, prettier

1. **Kiến thức về thư viện, framework (NestJs)**

* Cài đặt môi trường, runtime
* NestJS CLI
* Modules
* Controllers
* Route Guards
* Services
* DTOs
* Validation: class-validator và class-transformer.
* Middleware, Pipes, Guards và Interceptors
* Dependency Injection
* Connect database: TypeORM,  Entity
* Authentication và authorization
* Logging
* Notification với firebase
* Transactions
* CRUD
* Filter và pagination

1. **Kiến thức về cơ sở dữ liệu**

* Cài đặt môi trường
* Các khái niệm
* Các kiểu dữ liệu
* Các câu lệnh: Select, insert, update, delete
* Các thao tác: Tạo chỉnh sửa, xóa bảng
* Ràng buộc và chỉ mục
* Các phép toán và hàm trong MySQL
* Các mối quan hệ giữa các bảng (JOIN)
* Normalization và Denormalization (1NF, 2NF, 3NF)

1. **Kiến thức về web và API**

* HTTP và các phương thức HTTP: GET, POST, PUT, PATCH, DELETE
* Cấu trúc api: Response, request, body, header
* Status code
* RESTful APIs
* Json

**FRONTEND REACT**

**YÊU CẦU:**

- Có kiến thức cơ bản về HMTL, CSS và JAVASCRIPT

- HTML, CSS: Cấu trúc 1 trang web, Semantic HTML, Styling cơ bản (Flexbox, Grid, Responsive Design)

- JAVASCRIPT: Biến, kiểu dữ liệu, toán tử, Cấu trúc điều kiện, vòng lặp

- Biết sử dụng GIT và GITLAB để quản lý source code

**NỘI DUNG TỰ HỌC:**

**1. JAVASCRIPT và TYPESCRIPT**

- Javascript ES6 (Arrow function, Destructuring, Spread, Rest, Template Literals, Modules)

- Các kiểu dữ liệu trong typescript

- Interface và Type (so sánh, ứng dụng)

- Generics và Utility Types

**2. Làm quen với REACT và TYPESCRIPT**

- React Cơ bản

Cài đặt dự án: Vite với TypeScript.

+ JSX, Component, Props, State, Component lifecycle

+ Hooks: useState, useEffect.

+ Event handling trong React.

- TypeScript với React

+ Props vs State và định nghĩa props bằng interface/type.

+ Typing cho state và hooks.

+ Xử lý event với type (React.MouseEvent, React.ChangeEvent).

- Quản lý trạng thái & Gọi API & định tuyến trang web

+ Redux Toolkit (State Global)

+ Fetch API & Axios

+ React Router: Cài đặt, Route, Link, useParams, nested routes, Lazy loading với React.lazy và Suspense

**3. UI Libraries & Styling**

- Ant design (version 4)

- Styled-Components

**QC - Tester**

**I. Kiến thức về kiểm thử phần mềm (Software Testing)**

* **Kiến thức cơ bản về kiểm thử phần mềm**
  + Khái niệm kiểm thử phần mềm (Software Testing)
  + Mục đích và vai trò của kiểm thử phần mềm
  + Các loại kiểm thử: kiểm thử chức năng, kiểm thử phi chức năng
  + Phân biệt giữa kiểm thử manual và kiểm thử tự động
  + Các giai đoạn trong quy trình kiểm thử phần mềm: lập kế hoạch, thiết kế, thực hiện, báo cáo kết quả
* **Các loại kiểm thử**
  + **Kiểm thử chức năng (Functional Testing):** Kiểm thử tính năng phần mềm có hoạt động đúng như yêu cầu hay không.  
      
    - Kiểm thử đơn vị (Unit Testing)
    - Kiểm thử tích hợp (Integration Testing)
    - Kiểm thử hệ thống (System Testing)
    - Kiểm thử chấp nhận người dùng (User Acceptance Testing - UAT)
  + **Kiểm thử phi chức năng (Non-functional Testing):** Kiểm thử các yếu tố ngoài tính năng phần mềm như hiệu suất, bảo mật, khả năng sử dụng.  
      
    - Kiểm thử hiệu suất (Performance Testing)
    - Kiểm thử bảo mật (Security Testing)
    - Kiểm thử khả năng sử dụng (Usability Testing)
* **Các loại lỗi phần mềm:**
  + Lỗi chức năng, lỗi giao diện người dùng, lỗi hiệu suất, lỗi bảo mật, v.v.

**II. Công cụ kiểm thử và môi trường**

* **Công cụ quản lý kiểm thử:**
  + Jira, Trello, Asana (quản lý công việc kiểm thử)
  + TestRail (quản lý ca kiểm thử, test case)
  + Excel
* **Công cụ bug tracking:**
  + Bugzilla, Redmine, Jira (quản lý lỗi)
* **Công cụ kiểm thử API:**
  + Postman (kiểm thử API thủ công)
  + SoapUI (kiểm thử API SOAP)
* **Công cụ kiểm thử giao diện người dùng:**
  + Selenium (dành cho kiểm thử tự động, tuy nhiên có thể tham khảo cách thực hiện kiểm thử UI thủ công)
  + Browser Developer Tools (kiểm tra UI và các vấn đề liên quan)
* **Công cụ tạo báo cáo và tài liệu kiểm thử:**
  + Excel, Google Sheets (sử dụng để ghi chép kết quả kiểm thử)
  + TestLink (quản lý test case)

**III. Kiến thức về quy trình kiểm thử**

* **Lập kế hoạch kiểm thử:**
  + Xác định phạm vi kiểm thử
  + Xác định các yêu cầu kiểm thử
  + Thiết kế test case và test scenario
  + Lập kế hoạch kiểm thử (Test Plan)
* **Thực hiện kiểm thử:**
  + Thực hiện các test case
  + Ghi chép kết quả kiểm thử
  + Xử lý lỗi và thông báo lỗi cho nhóm phát triển
* **Báo cáo kết quả kiểm thử:**
  + Viết báo cáo kiểm thử (Test Summary Report)
  + Phân tích kết quả kiểm thử: Lỗi, tình trạng thành công, tiến độ kiểm thử

**IV. Kiến thức về kỹ thuật và phát triển phần mềm**

* **Hiểu biết về quy trình phát triển phần mềm:**
  + Waterfall, Agile (Scrum, Kanban)
  + Vòng đời phát triển phần mềm (SDLC - Software Development Life Cycle)
* **Kiến thức cơ bản về lập trình:**
  + Hiểu cơ bản về các ngôn ngữ lập trình (JavaScript, Python, Java, v.v.)
  + Các khái niệm về lập trình như biến, kiểu dữ liệu, hàm, điều kiện, vòng lặp (giúp hiểu được mã nguồn và thông báo lỗi)
* **Quản lý phiên bản phần mềm (Version Control):**
  + Git (cơ bản như commit, push, pull, merge)

**V. Kiến thức về Web và API**

* **Kiến thức về HTTP và API:**
  + Các phương thức HTTP (GET, POST, PUT, DELETE, PATCH)
  + Các mã trạng thái HTTP (200, 404, 500, v.v.)
  + Thực hiện kiểm thử API với Postman (các yêu cầu GET, POST, PUT, DELETE, kiểm tra response)
* **Kiểm thử RESTful API:**
  + Kiểm tra các điểm cuối (endpoints) của API
  + Kiểm tra dữ liệu trả về (response body)
  + Kiểm tra mã trạng thái HTTP (status codes)
* **Kiểm thử giao diện người dùng (UI Testing):**
  + Kiểm tra tính khả dụng của giao diện (usability)
  + Kiểm thử tương thích trình duyệt
  + Kiểm tra tính năng giao diện: form, button, menu, v.v.

**VI. Kiến thức về cơ sở dữ liệu**

* **Kiểm thử cơ sở dữ liệu:**
  + Kiểm tra tính toàn vẹn dữ liệu (data integrity)
  + Kiểm tra câu lệnh SQL: SELECT, INSERT, UPDATE, DELETE
  + Kiểm tra cơ sở dữ liệu quan hệ (JOIN, UNION, GROUP BY)
  + Kiểm tra các trigger và stored procedures
  + Kiểm thử với các công cụ như SQL Server, MySQL, PostgreSQL
* **Kiểm thử dữ liệu lớn (Data Testing):**
  + Xác nhận tính đúng đắn của dữ liệu
  + Kiểm tra dữ liệu đầu vào và đầu ra

**VII. Kiến thức về kiểm thử di động (Mobile Testing)**

* **Kiểm thử ứng dụng di động:**
  + Kiểm thử ứng dụng Android và iOS
  + Công cụ kiểm thử di động: Appium, TestFlight (iOS), Android Emulator, v.v.

**VIII. Các kỹ năng mềm**

* **Kỹ năng giao tiếp:**
  + Thảo luận về kết quả kiểm thử với nhóm phát triển
  + Viết báo cáo kiểm thử rõ ràng, chi tiết
  + Giao tiếp hiệu quả với các bên liên quan
* **Kỹ năng giải quyết vấn đề:**
  + Tìm ra nguyên nhân gốc rễ của lỗi
  + Đưa ra các giải pháp kiểm thử hợp lý