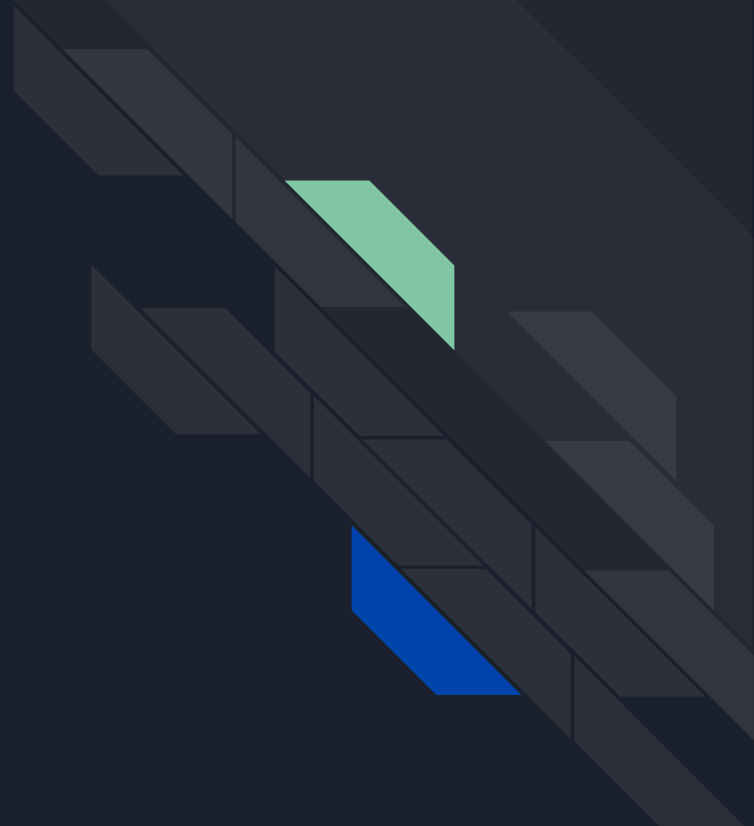


# CFD NÂNG CAO

1. Javascript nâng cao
2. React Js
3. Animation



# Chào! CFD TEAM

1. Ai học?
2. Học cái gì?
3. Học như thế nào?
4. Học rồi cuối cùng được cái gì?



# Đạt được sau khóa học

1. Ít nhất 2 dự án sau khi hoàn thành khóa học.
2. Nâng cao trình độ Javascript, ES6.
3. Có khả năng phân tích, thiết kế và phát triển sản phẩm sử dụng React JS
4. Biết cách sử dụng các thư viện đi cùng: react-router-dom, redux...
5. Hiểu về Ajax, RESTful API, GraphQL.
6. Biết cách sử dụng Git.
7. Biết cách sử dụng Grunt.
8. Tư duy về Animation, Gsap
9. Tự tin trả lời các câu hỏi phỏng vấn về React JS.

# Yêu cầu trước khi học

1. Có thể coding web responsive từ bản thiết kế (CFD căn bản).
2. Kiến thức cơ bản về Javascript, Javascript DOM.
3. Tinh thần ham học hỏi, chịu khó và tham gia đầy đủ các buổi học.



# Giới thiệu khóa học React JS

Ngày 1 - 3 - **CHUẨN BỊ**: Ôn tập kiến thức - Javascript nâng cao:

1. Cài đặt grunt và các module liên quan, cài đặt git, cấu trúc thư mục, cắt giao diện.
2. Javascript Nâng cao, ES6.
3. Promise, Async/Await, Try/Catch, JSON, Fetch, API.

Ngày 4 - 6 - **TẬP LUYỆN**: Làm quen với React cơ bản:

4. Làm quen với React: Component, cấu trúc thư mục, Props, state, Ref, Validate form, xử lý sự kiện.
5. Sử dụng Hook, làm quen với LifeCycle.
6. React nâng cao



# Giới thiệu khóa học Reactjs

Ngày 7 - 10 - **CHẠY**: React nâng cao - Sử dụng thư viện:

7. Cài đặt React-router-dom, xây dựng trang Single Page Application (SPA), các trường hợp thường gặp khi sử dụng.
8. Cài đặt Redux, giới thiệu và thực hành (Giới thiệu Context, Reducer).
9. Hướng dẫn thực hành RESTful API.
10. Thực hành React-router-dom và Redux.

Ngày 11 - 13 - **BỨT PHÁ**: Thực hành Shop bán hàng online:

11. Giới thiệu dự án, yêu cầu, hướng dẫn cắt giao diện
12. Hiện thị sản phẩm, đăng ký - đăng nhập, Giỏ hàng
13. Thanh toán, tìm kiếm sản phẩm



# Giới thiệu khóa học React JS

Ngày 14 - 15 - **LỢI NGƯỢC DÒNG**: Tư duy Animation, sử dụng thư viện Gsap:

14. Tư duy Animation
15. Gsap căn bản và các thư viện xử lý Animation.

Ngày 16 - 18 - **VỀ ĐÍCH**: Thực hành dự án cuối khóa:

16. Giới thiệu, yêu cầu và hướng dẫn dự án cuối khóa.
17. Thực hành dự án có giám sát.
18. Tối ưu dự án, build sản phẩm hoàn chỉnh.



**CHUẨN BỊ**



# CHUẨN BỊ

1. Cài đặt grunt và các module liên quan, cài đặt git, cấu trúc thư mục, cắt giao diện.
2. Javascript Nâng cao, ES6, ES7.
3. Promise, Async/Await, Try/Catch, Fetch, JSON, API.



1. Cài đặt grunt.
2. Cài đặt git.
3. Cấu trúc thư mục
4. Coding giao diện

# Grunt

1. Grunt là gì?
2. Cài đặt Grunt
3. Xây dựng cấu trúc thư mục project



# Grunt là gì?

- ❖ Grunt là 1 công cụ để build code cũng như tự động các task lặp đi lặp lại (task runner). Công việc của task runner như nén, biên dịch các file scss, javascript, live server v.v.. nó được viết bằng Node.js.
- ❖ Hệ sinh thái Grunt rất lớn và phong phú.
- ❖ Số lượng plugin lớn, hỗ trợ tốt việc chạy tự động các task.
- ❖ Nếu muốn sử dụng được nhiều plugin hơn được hỗ trợ tốt hơn nên dùng **Grunt** còn nếu dự án cần ưu tiên tốc độ xử lý hơn và không yêu cầu quá nhiều các thư viện phức tạp hãy chọn **Gulp**.



# Cài đặt Grunt

1. **Cài đặt Node JS:** <https://nodejs.org/en/download/>
2. **Cài đặt Ruby:** <https://rubyinstaller.org/downloads/>
3. **Cài đặt Grunt CLI:** `npm install grunt-cli -g`
4. **Tạo thư mục dự án và di chuyển vào:** `cd duong-dan-thu-muc`
5. **Khởi tạo thông tin project:** `npm init`
6. **Cài đặt plugin và module:** `npm install grunt --save-dev`
7. **Tạo file Gruntfile.js để biết cấu hình**



# Cài đặt Plugin Grunt

1. **Tạo thư mục dự án và chạy CMD**
2. **loadNpmTasks:** `npm install --save-dev load-grunt-tasks`
3. **CSSMin:** `npm install grunt-contrib-cssmin --save-dev`
4. **Uglify:** `npm install grunt-contrib-uglify --save-dev`
5. **Di chuyển vào C:\Windows\System32 chạy lệnh:** `Gem install sass` **chọn 1 enter.**
6. **SCSS:** `npm install grunt-contrib-sass --save-dev`
7. **Watch tự động hóa:** `npm install grunt-contrib-watch --save-dev`
8. **BrowserSync:** `npm install grunt-browser-sync --save-dev`

# Tổ chức thư mục

Hướng dẫn tổ chức và cấu trúc  
thư mục dự án.

# Git

1. Git là gì?
2. Cài đặt Git
3. Các câu lệnh thường gặp trong git





# Git là gì?

- ❖ Git là một "hệ thống điều khiển phiên bản phân tán" (Distributed version control system) miễn phí và mã nguồn mở, nó được thiết kế để một nhóm làm việc cùng nhau có thể chia sẻ và quản lý phiên bản các tập tin trong quá trình làm việc.
- ❖ Tiết kiệm thời gian
- ❖ Làm việc offline
- ❖ Khôi phục khi gặp lỗi, yên tâm
- ❖ Không trộn lẫn các công việc
- ❖ Cộng đồng người dùng lớn

# Cài đặt Git

Tham khảo:

<https://kipalog.com/posts/Huong-dan-chi-tiet-tung-buoc-cai-dat-Git-cho-Windows>

# Các câu lệnh thường gặp trong git

Tham khảo:

<https://viblo.asia/p/tap-hop-nhung-cau-lenh-git-huu-dung-dWrwwWr2vw38>



# Câu lệnh thường gặp trong git

- ❖ **Git config:**

```
git config --global user.name "John Doe"  
git config --global user.email "john@example.com"
```

- ❖ **Giúp Git bỏ qua folder node\_modules:**

Tạo file `.gitignore` rồi thêm vào nội dung:

```
# dependencies  
/node_modules
```

- ❖ **Khởi tạo Git repo cho code có sẵn:**

```
cd existing-project/  
git init
```

- ❖ **Clone một remote repo:** `git clone https://github.com/user/repository.git`



# Câu lệnh thường gặp trong git

- ❖ **Cập nhật thay đổi:**  
git add .  
git commit -m "Message"
- ❖ **Cập nhật lên server:**  
git push origin <name\_branch>
- ❖ **Xem lại lịch sử commit:** git log
- ❖ **Xem thay đổi trước khi push:** git diff
- ❖ **Pull từ remote repository:** git pull origin master



# Các câu hỏi thường gặp khi phỏng vấn

1. Git fork là gì? Sự khác nhau giữa git fork, branch và clone?
2. Sự khác nhau giữa pull request và branch?
3. Sự khác nhau giữa git pull và git fetch?
4. Làm thế nào để revert previous commit trong git?
5. git cherry-pick là gì?
6. Giải thích những ưu điểm of Forking Workflow?
7. Sự khác nhau giữa HEAD, working tree và index?
8. Trình bày quy trình làm việc của Gitflow Workflow?
9. Khi nào nên sử dụng git stash?
10. Làm thế nào để loại bỏ một tập tin từ git mà không cần loại bỏ nó khỏi file system của bạn?
11. Khi nào nên sử dụng git rebase thay vì git merge?

# Thực hành

1. Cắt giao diện website CFD
2. Đưa lên Github