

# Fullstack Resource and guideline

## ▼ List of things to Learn to become a Full stack Engineer

### List of things to Learn to become a Full stack Engineer

- learn Git
  - create a GitHub account
  - learn how to commit and save code locally and remotely
  - learn how to merge other's code into yours
  - learn how to create a pull request to get feedback on your changes
- learn JavaScript
  - learn loops, control statements, data structures like array and object
  - learn asynchronous programming, Promises, async/await, the event loop- learn internet fundamentals
- learn HTTP vs HTTPS
- learn HTML and CSS
- learn REST standards. Especially methods GET, POST, PUT, and DELETE
- learn DNS and domains- learn a backend framework
- learn NodeJS
- learn about ExpressJS servers and middleware stacking
- learn about ORMs (Mongoose is one of my favorites) and how to get a server to talk to a database
- learn about server techniques to mitigate XSS, CSRF and SQL injection attacks
- check out GraphQL paradigm vs REST paradigm on the server- learn a front-end framework
- implement a simple ReactJS app
- learn about component design and higher-order components
- learn React hooks for state change management
- learn NextJS to try out server-side rendering- optimize your code
- learn data structures: stacks, queues, lists, maps and trees
- learn algorithms: breadth-first search, depth-first search, binary search, recursion
- learn TypeScript to make your code even better

## ▼ Some useful topics to discuss about javascript, fuctional programming and react

### Some useful topics to discuss about javascript, fuctional programming and react

- Components
- Functional programming
- Mathematical Function
- Difference between Object oriented programming & functional programming
- pass by value vs pass by reference
- passing function as a reference/arguments/props
- javascript function as first class citizen/object
- higher order function
- callback function
- JavaScript map, reduce, filter, forEach function
- method/function chaining
- higher order components
- middleware as cross cutting concerns
- Javascript context, object binding sharing
- problem in sharing state between highly nested components
- state management problem in vanilla javascript DOM
- react component rendering in state change

## ▼ Some useful topics to discuss about react hook and Jwt authentications

### Some useful topics to discuss about react hook and Jwt authentications

- Difference between reduce() & useReducer() hook
- useReducer() hook
- purpose of using useReducer() hook
- difference between useState() & useReducer()
- useReducer() for complex data & action handling
- useContext() hook for global storage management
- how to combine useReducer() and useContext() hook for global complex state management
- useEffect() hook & useReducer() hook
- JWT authentication token management
- keeping data in localStorage
- refactoring current context code in reducer, action, action type & creator file

- JWT token handling code refactoring from backend
- data & action management using useReducer() hook

#### ▼ Mern Stack Resource, topics and assignment

##### **Mern Stack Resource, topics and assignment**

##### **Resource**

[https://github.com/bradtraversy/mern\\_shopping\\_list](https://github.com/bradtraversy/mern_shopping_list)  
<https://www.youtube.com/playlist?list=PLlIGF-RfqbbiTGgA77tGO426V3hRF9iE>  
<https://www.youtube.com/playlist?list=PLC3y8-rFHvwwg3vaYJgHGnModB54rxOk3> (58 ~ 76)  
<https://github.com/gopinav/React-Tutorials/tree/master/React%20Hooks>  
<https://www.vivasoftltd.com/understanding-redux>  
<https://react-redux.js.org/tutorials/quick-start>

##### **Topics**

Redux(predictable state management(context) system for javascript applications)  
 React-redux(actions, reducers, store, connect, provides)  
 Redux-toolkit(useDispatch, useSelector)  
 useContext()  
 useReducer()  
 localStorage()  
 where & when to use global storage(redux or context)  
 problem using global storage(redux/context) for all kind of state handling  
 how to deal with nested pages/sections/components dependent state  
 middleware & private routing  
 JWT token handling  
 working in MERN stack

##### **Assignment**

[https://github.com/bradtraversy/mern\\_shopping\\_list](https://github.com/bradtraversy/mern_shopping_list)  
 rewrite this source code using next-js, useContext with useReducers & Redux-toolkit

#### ▼ JWT token as httpOnly cookie & Next.Js version >= 12.0 middleware

##### **JWT token as httpOnly cookie & Next.Js version >= 12.0 middleware**

<https://www.youtube.com/watch?v=T6fRWZWvJzI>  
<https://www.youtube.com/watch?v=DHZSYTCTbA>  
<https://www.youtube.com/watch?v=w8n7Soz7khw>

##### **Next Auth:**

<https://www.youtube.com/watch?v=AI9jExb5dDA>  
<https://www.hamedbahram.io/notes/next-auth-v4>

##### **Create custom page in next-auth**

[https://www.youtube.com/watch?v=kOXb\\_NSBZrw](https://www.youtube.com/watch?v=kOXb_NSBZrw)

✓ Goldmine for UI Designer

### **Goldmine for UI Designer**

[https://www.linkedin.com/posts/bikashjoshi\\_colour-typography-layouts-activity-7002498641634430976-84wZ](https://www.linkedin.com/posts/bikashjoshi_colour-typography-layouts-activity-7002498641634430976-84wZ)

I have listed some of the most common and important principle in UI Design and have also mentioned what to specifically learn in each one of this principle along with the popular blogs to help you get started.

#### **Principle 1: #Colour**

Learn the history of colour:

<https://lnkd.in/dCF4daw8>

Learn the colour psychology:

<https://lnkd.in/d3RfTbz>

Learn the science behind how do we differentiate between colours.

<https://lnkd.in/d-45yEY3>

Learn the difference between RGB, CMYK, CMYKOGV, AND SPOT COLORS

<https://lnkd.in/dkaEXbjm>

#### **Principle 2: #Typography**

Learn the history of typography: e.g., learn when and how did we started to use typefaces and how did the typeface evolve from printing to digital.

[https://lnkd.in/dPhdN\\_3k](https://lnkd.in/dPhdN_3k)

Learn about the typeface psychology: e.g., why does serif typefaces gives us the feeling of trust.

<https://lnkd.in/dnzfu9EW>

Learn about how to choose typefaces in your design work.

<https://lnkd.in/dsjrYAwP>

Principle 3: #Layouts and #Composition

<https://lnkd.in/dUyRybZd>

<https://lnkd.in/dBupcG8E>

#### **Principle 4: #WhiteSpace or #NegativeSpace**

Learn the importance of white space and how to effective use white space to improvise your design

<https://lnkd.in/dpza-93T>

<https://lnkd.in/du85FRkV>

#### **Principle 5: #Consistency**

Learn what is consistency, why it is important and how to have consistency in design

<https://lnkd.in/dKV-72fq>

#### **Principle 6: #GridSystem**

Learn how to apply grid system and how we should align our design components in the grid

<https://lnkd.in/d5EmB88U>

All the best.

