

Meta Front-End Developer Professional Certificate

Tools and software

- ❖ HTML-HTML5
- ❖ CSS
- ❖ Bootstrap
- ❖ JavaScript
- ❖ React
- ❖ React.js
- ❖ JSX
- ❖ JEST & React Testing Library
- ❖ Node.js
- ❖ NPM
- ❖ VSCode
- ❖ GitHub
- ❖ Git
- ❖ Bash (Unix Shell)
- ❖ Figma
- ❖ Live Server
- ❖ Browser Preview
- ❖ CodeRunner
- ❖ Stylelint
- ❖ ChakraUI
- ❖ Formik
- ❖ Yup resolver
- ❖ Netlify

Skills

- ❖ Web Development/Tools
- ❖ Web Application
- ❖ Application development
- ❖ User Interface
- ❖ Front-End Web Development
- ❖ HTML and CSS
- ❖ Responsive Web Design
- ❖ Test-Driven Development
- ❖ JavaScript
- ❖ Object-Oriented Programming (OOP)
- ❖ Linux
- ❖ Bash (Unix Shell)
- ❖ Github
- ❖ Version Control
- ❖ Debugging
- ❖ React (Web Framework)
- ❖ Unit Testing
- ❖ Computer Programming
- ❖ Computer Interaction
- ❖ User Experience
- ❖ Interactive Design

HTML

- ❖ Layout & Style
- ❖ Text formatting & lists
- ❖ Images and links
- ❖ Linking and Meta
- ❖ Semantic

CSS

- ❖ Font styling (font size, font color, etc.)
- ❖ Flex Box Layout (Layout of items using CSS Flex Box Layout)
- ❖ CSS Selectors
- ❖ Position, Padding, Margins and Display
- ❖ Colors, Background and Icons

JavaScript

- ❖ Data types
- ❖ Using var, let and const
- ❖ Conditionals and Loops
- ❖ Using objects, arrays and functions
- ❖ ES6 Arrow functions
- ❖ In-built functions such as map(), forEach() and promises.
- ❖ Destructuring Arrays and Objects
- ❖ Error Handling
- ❖ Package Manager (Node + npm)

React Basics

- ❖ Functional components
- ❖ React components
- ❖ Transpile JSX
- ❖ Project Structure
- ❖ Create/Import Components
- ❖ Props
- ❖ JSX Expressions
- ❖ Multiple Components
- ❖ Dynamic events
- ❖ Data Flow
- ❖ Hooks
- ❖ State
- ❖ Prop Drilling
- ❖ State/Stateless
- ❖ Managing state
- ❖ Routing
- ❖ Assets

Advanced React

- ❖ Rendering List
- ❖ Keys
- ❖ Forms
- ❖ Controlled components vs. Uncontrolled components
- ❖ React Context
- ❖ Hooks (useState, useEffect, useRef)
- ❖ Custom Hooks
- ❖ JSX
- ❖ Components vs. Elements
- ❖ React Declarative Model
- ❖ Performance
- ❖ Components Composition
- ❖ Containment vs. Specialization
- ❖ Types of children in JSX
 - string literals
 - jsx elements
 - js expressions
 - functions
 - boolean, null and undefined
- ❖ React API top level
 - React.cloneElements
 - React.children
- ❖ Spread Operator
- ❖ Cross cutting concerns
- ❖ Higher order component (HOC)
- ❖ Reusing behaviour
- ❖ React Testing Library
- ❖ JEST
- ❖ CI (Continuous integration)
- ❖ Style Guides
- ❖ ChakraUI
- ❖ Formik
- ❖ Yup resolver
- ❖ Netlify

Principles of UX/UI Design

- ❖ Fundamentals of User Experience (UX)
- ❖ Accessibility considerations in design
- ❖ Developing user empathy
- ❖ Create wireframes and prototypes in Figma
- ❖ UX artifacts such as user personas
- ❖ Minimum Viable Product (MVP) style prototypes
- ❖ UX UI development
- ❖ Evaluating interactive design
- ❖ Consistency and quality
- ❖ Frames, layers and basic shapes
- ❖ Groups
- ❖ Type and Text

- ❖ Typography
- ❖ Grids and constraints
- ❖ Grid systems
- ❖ Manipulating elements
- ❖ Images
- ❖ Wireframing
- ❖ Rapid Prototyping
- ❖ Usability testing
- ❖ High-fidelity prototype
- ❖ High-fidelity user interface elements
- ❖ Design systems
- ❖ Advanced UI techniques in Figma
- ❖ Design Elements Principles
- ❖ Stylize High-fidelity design
- ❖ Design to prototyping
- ❖ Micro-interactions and micro-animations
- ❖ High-fidelity design prototype

Capstone Project

- ❖ React
- ❖ Figma
- ❖ Git
- ❖ GitHub
- ❖ VS Code
- ❖ Netlify

Coding Interview Preparation

- ❖ What is a coding interview?
- ❖ Types of interviews
- ❖ Preparation
- ❖ Communication
- ❖ Pseudo code
- ❖ Tips
- ❖ Solution testing
- ❖ Binary
- ❖ RAM: Read Access Memory
- ❖ ROM: Read Only Memory
- ❖ Big-O Notation: Determine an algorithm's efficiency
- ❖ Notation:
 - $O(1)$
 - $O(\log \log n)$
 - $O(\log n)$
 - $O(n)$
 - $O(n \log n)$
 - $O(n^2)$
 - $O(n^3)$
 - $O(2n)$
- ❖ Space complexity = input space + auxiliary space

❖ Data Structure

- Basics:
 - Strings
 - Integers
 - Boolean
 - Arrays
 - Objects
 - Lists/Sets
 - Stacks/Queues/Trees
- Advanced:
 - Hash tables
 - Heaps
 - Graphs

❖ Algorithms

- Sorting: Selection, insertion and Quick
- Searching
- Time and space complexity
- Divide and conquer
- Recursion
- Dynamic programming
- Greedy algorithm's

Next features: focus on Leet Code and Mock interviews (practice).

Commands

```
$ npm config set registry http://registry.npmjs.org/  
$ npm init react-app firstapp  
$ npm install
```