**MERN STACK CAREER ROADMAP**

**Summery**

1. **Fundamental Web Development Skills:**
   * HTML/CSS: Learn the basics of HTML and CSS for creating the structure and styling of web pages.
   * JavaScript: Master JavaScript, as it's the foundation for both front-end and back-end development.
2. Front-End Development:
   * React: Start by learning React, a popular JavaScript library for building user interfaces.
     + Component-based architecture
     + State management (React hooks, Redux)
     + Routing (React Router)
     + Asynchronous data fetching (Axios, Fetch API)
3. Back-End Development:
   * Node.js: Learn Node.js for server-side development.
     + Asynchronous JavaScript (Promises, Async/Await)
     + Building RESTful APIs with Express.js
     + Authentication and authorization (JWT, Passport.js)
     + Database integration (MongoDB/Mongoose for NoSQL databases)
4. Databases:
   * MongoDB: Understand NoSQL databases and how to interact with them using MongoDB.
     + Data modeling
     + CRUD operations
     + Indexing and optimization
5. Version Control:
   * Learn Git and GitHub to manage your code and collaborate with others.
6. API Development:
   * Build RESTful APIs using Express.js.
   * Learn about GraphQL if needed for specific projects.
7. State Management:
   * Understand state management solutions like Redux or MobX for complex React applications.
8. Authentication and Security:
   * Implement user authentication and authorization.
   * Learn about common security vulnerabilities (e.g., Cross-Site Scripting, SQL Injection) and how to mitigate them.
9. Testing:
   * Write unit tests using testing libraries like Jest for both front-end and back-end code.
10. Deployment:
    * Deploy your MERN stack applications to cloud platforms like Heroku, or Netlify.
    * Learn about containerization with Docker.
11. Performance Optimization:
    * Optimize front-end and back-end code for performance.
    * Implement caching strategies.
    * Monitor and troubleshoot performance issues.
12. Continuous Integration and Deployment (CI/CD):
    * Set up automated build and deployment pipelines with tools like Jenkins, Travis CI, or GitHub Actions.
13. Additional Skills:
    * Learn about WebSockets for real-time features.
    * Explore serverless architecture with AWS Lambda or similar services.
    * Familiarize yourself with other MERN stack alternatives like MEAN (MongoDB, Express.js, Angular, Node.js) or JAMstack (JavaScript, APIs, and Markup).
14. Project Portfolio:
    * Build a portfolio of projects showcasing your MERN stack development skills.
    * Include personal projects and contributions to open-source projects if possible.
15. Stay Updated:
    * Keep up with the latest trends and updates in the MERN stack ecosystem by following blogs, forums, and online communities.
16. Networking:
    * Attend meetups, conferences, and join developer communities to network with other professionals in the field.
17. Soft Skills:
    * Improve your communication and teamwork skills, as software development often involves collaboration.
18. Job Search:
    * Prepare a strong resume and cover letter.
    * Apply for MERN stack developer positions, and be prepared for technical interviews.

**Details**

* HTML
  1. HTML Introduction
  2. HTML Elements
  3. HTML Attributes
  4. HTML Headings
  5. HTML Paragraphs
  6. HTML Styles
  7. HTML Formating
  8. HTML Quotations
  9. HTML Comments
  10. HTML Colors
  11. HTML CSS
  12. HTML Links
  13. HTML Images
  14. HTML Favicon
  15. HTML Page Title
  16. HTML Tables
  17. HTML Lists
  18. HTML Block & Inline
  19. HTML Classes
  20. HTML Id
  21. HTML Iframes
  22. HTML JavaScript
  23. HTML File Paths
  24. HTML Head
  25. HTML Layout
  26. HTML Responsive
  27. HTML Computercode
  28. HTML Semantics
  29. HTML Style Guide
  30. HTML Entities
  31. HTML Symbols
  32. HTML Emojis
  33. HTML Charset
  34. HTML URL Encode
  35. HTML Forms
  36. HTML Canvas
  37. HTML SVG
  38. HTML Media
  39. HTML Video
  40. HTML Audio
  41. HTML Plug-ins
  42. HTML Youtube
* CSS

1. CSS Introduction
2. CSS Syntax
3. CSS Selectors
4. CSS How To
5. CSS Comments
6. CSS Colors
7. CSS Backgrounds
8. CSS Borders
9. CSS Margins
10. CSS Padding
11. CSS Height/Weight
12. CSS Box Model
13. CSS Outline
14. CSS Text
15. CSS Fonts
16. CSS Icons
17. CSS Links
18. CSS Lists
19. CSS Tables
20. CSS Display
21. CSS Max-Width
22. CSS Position
23. CSS Z-index
24. CSS Overflow
25. CSS Float
26. CSS Inline-block
27. CSS Align
28. CSS Combinators
29. CSS Pseudo-class
30. CSS Pseudo-element
31. CSS Opacity
32. CSS Navigation Bar
33. CSS Dropdowns
34. CSS Image Gallery
35. CSS Image Sprities
36. CSS Attr Selectors
37. CSS Forms
38. CSS Couters
39. CSS Website Layout
40. CSS Units
41. CSS Specificity
42. CSS Important
43. CSS Math Functions
44. CSS Rounded Corners
45. CSS Border Images
46. CSS Backgrounds
47. CSS Colors
48. CSS Color Keywords
49. CSS Gradients
50. CSS Shadows
51. CSS Text Effects
52. CSS Web Fonts
53. CSS 2D Transforms
54. CSS 3D Transforms
55. CSS Transitions
56. CSS Animations
57. CSS Tooltips
58. CSS Style Images
59. CSS Image Reflection
60. CSS object-fit
61. CSS object-position
62. CSS Masking
63. CSS Buttons
64. CSS Pagination
65. CSS Multiple Columns
66. CSS User Interface
67. CSS Variables
68. CSS Box Sizing
69. CSS Media Queries
70. CSS Flexbox
71. RWD Intro
72. RWD Viewport
73. RWD Grid View
74. RWD Media Queries
75. RWD Images
76. RWD Videos
77. RWD Frameworks
78. RWD Templates
79. Grid Intro
80. Grid Container
81. Grid Item
82. SASS

* Bootstrap
  1. BS Get Started
  2. BS Containers
  3. BS Grid Basic
  4. BS Typography
  5. BS Colors
  6. BS Tables
  7. BS Images
  8. BS Jumbotron
  9. BS Alerts
  10. BS Buttons
  11. BS Button Groups
  12. Bootstrap Badges
  13. BS Progress Bars
  14. BS Spinners
  15. BS Pagination
  16. BS List Groups
  17. BS Cards
  18. BS Dropdowns
  19. BS Collapse
  20. BS Navs
  21. BS Navbar
  22. BS Carousel
  23. BS Modal
  24. BS Tooltip
  25. BS Popover
  26. BS Toast
  27. BS Scrollspy
  28. BS Offcanvas
  29. BS Utilities
  30. BS Dark Mode
  31. BS Flex
  32. BS Forms
  33. BS Grid
* JavaScript
  1. JS Introduction
  2. JS Where To
  3. JS Output
  4. JS Statements
  5. JS Syntax
  6. JS Comments
  7. JS Variables
  8. JS Let
  9. JS Const
  10. JS Operators
  11. JS Arithmetic
  12. JS Assignment
  13. JS Data Types
  14. JS Functions
  15. JS Objects
  16. JS Events
  17. JS Strings
  18. JS String Method
  19. JS String Search
  20. JS String Templates
  21. JS Numbers
  22. JS Bight
  23. JS Number Mehtods
  24. JS Number Properties
  25. JS Array
  26. JS Array Methods
  27. JS Array Sort
  28. JS Array Iteration
  29. JS Array Const
  30. JS Dates
  31. JS Date Formates
  32. JS Date Get Methods
  33. JS Date Set Methods
  34. JS Math
  35. JS Random
  36. JS Booleans
  37. JS Comprisons
  38. JS If Else
  39. JS Switch
  40. JS Loop For
  41. JS Loop For In
  42. JS Loop For Of
  43. JS Loop While
  44. JS Break
  45. JS Iterables
  46. JS Sets
  47. JS Maps
  48. JS Typeof
  49. JS Type Conversion
  50. JS Bitwise
  51. JS RegExp
  52. JS Presedence
  53. JS Errors
  54. JS Scope
  55. JS Hoisting
  56. JS Strict Mode
  57. JS this Keyword
  58. JS Arrow Funcion
  59. JS Classes
  60. JS Modules
  61. JS JSON
  62. JS Debugging
  63. JS Best Practices
  64. JS Style Guide
  65. JS Mistakes
  66. JS Performance
  67. JS Reserved Words
  68. JS ES6 Versions
  69. Object Definition
  70. Object Properties
  71. Object Mehods
  72. Object Display
  73. Object Accessors
  74. Object Constructors
  75. Object Prototypes
  76. Object Iterables
  77. Object Sets
  78. Object Maps
  79. Function Definitons
  80. Function Parameters
  81. Function Invocation
  82. Function Call
  83. Function Apply
  84. Function Bind
  85. Function Clousers
  86. Class Intro
  87. Class Inheritence
  88. Class Static
  89. JS Callbacks
  90. JS Asynchronous
  91. JS Promise
  92. JS Async/Await
  93. DOM Intro
  94. DOM Methods
  95. DOM Documents
  96. DOM Elements
  97. DOM HTML
  98. DOM Forms
  99. DOM CSS
  100. DOM Animations
  101. DOM Events
  102. DOM Event Listener
  103. DOM Navigation
  104. DOM Nodes
  105. DOM Collection
  106. DOM Node List
  107. BOM Window
  108. BOM Screen
  109. BOM Location
  110. BOM History
  111. BOM Navigator
  112. BOM Popup Alert
  113. BOM Timing
  114. BOM Cookies
  115. Web API Intro
  116. Web Forms API
  117. Web History API
  118. Web Storage API
  119. Web Worker API
  120. Web Fetch API
  121. Web Geolocation API
  122. AJAX Intro
  123. AJAX XML HTTP
  124. AJAX Request
  125. AJAX Response
  126. AJAX XML File
  127. JSON Intro
  128. JSON Syntax
  129. JSON vs XML
  130. JSON Data Types
  131. JSON Parse
  132. JSON Stringify
  133. JSON Objects
  134. JSON Arrays
  135. JSON Server
* React.js
  1. React Intro
  2. React Get Started
  3. React Upgrade
  4. React ES6
  5. React Render HTML
  6. React JSX
  7. React Components
  8. React Class
  9. React Props
  10. React Events
  11. React Conditionals
  12. React Lists
  13. React Forms
  14. React Router
  15. React Memo
  16. React CSS Styling
  17. React Sass Styling
  18. React Hooks Introduction
  19. React useState Hook
  20. React useEffect Hook
  21. React useContect Hook
  22. React useRef Hook
  23. React useReducer Hook
  24. React useCallback Hook
  25. React useMemo Hook
  26. React custom Hook
* Next.js
  1. React.js
  2. Set up
  3. Project Structure
  4. Routing
  5. Layouts and Components
  6. Styling
  7. Data Fetching
  8. State Management
  9. Authentication and Authorization
  10. SEO Optimization
* Node.js ……………………………Loading
  1. Node History,Story *…………………………….finished*
  2. *2009 – node.js*
  3. *2010 – npm*
  4. *2010 – mongodb*
  5. *json*
  6. *react.js*
  7. *express.js*
  8. *then popular grouwing node*
  9. *javascript everywhere*
  10. *node on mobile*
  11. *node on desktop*
  12. Node How Work? *………………………………...finished*
  13. *runtime*
  14. *v8 engine*
  15. *threads*
  16. *server*
  17. *database*
  18. *single thread*
  19. *request/response*
  20. *assistent/worker*
  21. *no-blocking i/o*
  22. *asynchronous*
  23. *callback*
  24. *cpu intensive*
  25. *event loop*
  26. *libuv*
  27. *kernel*
  28. *multi-thread*
  29. *then blessing fast, high performance*
  30. Node Intro………………………………*finished*
  31. Node Get Started………………………*.finished*
  32. Node Modules System*…………………………..finished*
  33. *encapsulated*
  34. *wrapper function*
  35. *exports // named exportts*
  36. *module.exports // default exports*
  37. *require() // do not need .js*
  38. *\_\_dirname*
  39. *\_\_filename*
  40. *“type”: “module” // ES6 module*
  41. *import, export // must be use .js*
  42. *born npm*
  43. *three types module – external,custom,default*
  44. Node Global Object *……………………………..finished*
  45. *not have window*
  46. *global*
  47. *setTimeout()*
  48. *setInterval()*
  49. *clearTimeout()*
  50. *clearInterval()*
  51. *fetch()*
  52. *clearImmediate()*
  53. *setImmediate()*
  54. *queueMicrotask()*
  55. Node Include Modules*………………...finished*
  56. *require()*
  57. Node Create Custom Modules*…………...finished*
  58. *exports.custom = custom*
  59. Node Path Module *………………………………………...finished*
  60. *path.basename()*
  61. *path.dirname()*
  62. *path.extname()*
  63. *path.format()*
  64. *path.parse()*
  65. Node Os Module *…………………………………..finished*
  66. *os.cpus()*
  67. *os.freemem()*
  68. *os.homedir()*
  69. *os.hostname()*
  70. *os.platform()*
  71. *os.tmpdir()*
  72. *os.version()*
  73. Node HTTP Modules*………….finished*
  74. *http.createServer().listen()*
  75. *http.createServer().on()*
  76. *req.url*
  77. *req.method*
  78. *res.writeHead()*
  79. *res.write()*
  80. *res.end()*
  81. Node File System*……………….finished*
  82. *require(‘fs’)*
  83. *fs.writeFile()*
  84. *fs.appendFile()*
  85. *fs.readFile()*
  86. *fs.unlink()*
  87. *fs.rename()*
  88. Node URL Module*…………………..finished*
  89. *require(‘url’)*
  90. *url.parse()*
  91. *url.format()*
  92. Node NPM*…………………………………..finished*
  93. *npm (Node Package Manager) was included with Node.js by default. When you installed Node.js, npm was automatically installed along with it. This allowed you to use npm to easily manage packages and dependencies for your Node.js projects.*
  94. *npm CLI*
  95. *See Node\_Package\_Manager.docx*
  96. Node Events*…………………………….finished*
  97. *require(‘events’)*
  98. *new events.EventEmmitter()*
  99. *eventEmmitter.on()*
  100. *eventEmmitter.emit()*
  101. *event bell ring module system*
  102. Node Upload File
  103. Node Email
  104. MySQL Get Started
  105. MySQL Create Database
  106. MySQL Create Table
  107. MySQL Insert Into
  108. MySQL Select From
  109. MySQL Where
  110. MySQL Order By
  111. MySQL Delete
  112. MySQL Drop Table
  113. MySQL Update
  114. MySQL Limit
  115. MySQL Join
  116. MongoDB Get Started
  117. MongoDB Create Database
  118. MongoDB Create Collection
  119. MongoDB Insert
  120. MongoDB Find
  121. MongoDB Query
  122. MongoDB Sort
  123. MongoDB Delete
  124. MongoDB Drop Collection
  125. MongoDB Update
  126. MongoDB Limit
  127. MongoDB Join
  128. Uses for Create Server ………………………………………loading
  129. Uses for request, response smooth handling ………………loading
  130. Uses for build scalable and high performance web application ………………..loading
  131. Uses for API Development …………………………loading
  132. Uses for create CLIs
  133. Uses for automate repettitive tasks
  134. Uses for system administration
  135. Uses for real-time application
  136. Uses for IoT application
  137. Uses for microservice application
  138. Uses for streamming services
  139. Uses for database operation
  140. Uses for authentication and authorization
  141. Uses for file system operation
  142. Uses for testing and automation
  143. Uses for package management
  144. Uses for dashboard an Reporting
  145. Uses for develop network applicaton
  146. Uses for create TCP/UDP server and client
  147. Uses for data processing
* Express.js

1. express()
2. express.json()
3. express.raw()
4. express.router()
5. express.static()
6. express.text()
7. express.urlencoded()
8. app.locals
9. app.mountpath
10. app.mount
11. app.all()
12. app.delete()
13. app.disable()
14. app.enable()
15. app.enabled()
16. app.engine()
17. app.get()
18. app.listen()
19. app.METHOD()
20. app.param()
21. app.path()
22. app.push()
23. app.put()
24. app.render()
25. app.route()
26. app.render()
27. app.set()
28. app.use()
29. req.app
30. req.baseUrl
31. req.body
32. req.cookies
33. req.fresh
34. req.hostname
35. req.ip
36. req.ips
37. req.mehod
38. req.originalUrl
39. req.params
40. req.path
41. req.protocol
42. req.query
43. req.route
44. req.secure
45. req.signedCookies
46. req.stale
47. req.subdomains
48. req.xhr
49. req.accepts()
50. req.accepsCharsets()
51. req.acceptsEncodings()
52. req.acceptsLanguages()
53. reg.get()
54. req.is()
55. req.param()
56. req.range()
57. res.app
58. res.headersSent
59. res.locals
60. res.append()
61. res.attachment()
62. res.cookie()
63. res.clearCookie()
64. res.download()
65. res.end()
66. res.format()
67. res.get()
68. res.json()
69. res.jsonp()
70. res.links()
71. res.location()
72. res.redirect()
73. res.render()
74. res.send()
75. res.sendFile()
76. res.set()
77. res.status()
78. res.type()
79. res.vary()
80. router.all()
81. router.METHOD()
82. router.route()
83. router.param()
84. router.use()

* MongoDB
  1. MongoDB Get Started
  2. MongoDB Query API
  3. MongoDB Create Database
  4. MongoDB Create Collection
  5. MongoDB Insert
  6. MongoDB Find
  7. MongoDB Update
  8. MongoDB Delete
  9. MongoDB Query Operators
  10. MongoDB Update Operators
  11. MongoDB Aggregations
  12. MongoDB Indexing/Search
  13. MongoDB Validation
  14. MongoDB Data API
  15. MongoDB Drivers
  16. MongoDB Node.js Drivers
  17. MongoDB Charts
* Project Portfolio
  1. A Ecommarce Project with Link
  2. A Inventory Manage Project With LInk
  3. A Dashboard Monitoring Project
  4. use react.js
  5. use socket.io
  6. use node.js
  7. use express.js
  8. use mongodb
  9. use mysql
  10. use bootstrap
  11. use tailwind
  12. use metarial
* My Portfolio
  1. jagadish.netlify.app
  2. use next.js
  3. use node.js
  4. use express.js
  5. use mongodb
  6. use metarial ui
  7. use tailwind