

MERN Stack Course Details

MindRisers!@

Environment Setup:

- **install node**

- Introduction to development tools
- Text Editor, IDE

- **Git**

Download from ([download git](#))

“ [also create an account in github.com](#) ”

- Git workflow(branching,push,pull)
Pull code from github to our device

1. git init // this will create a .git folder in our local project directory

2. git remote add origin <remote_remote_url>

// eg: <https://github.com/sagartmg2/express>

// to check , git remote -v

3. git pull origin <branch_name>

// eg : git pull origin master

// another process to pull our code

1.git clone <git-url>

//check our changes.

git status

// To specify which files to be uploaded to remote github repo

git add <file_name> // add a single file

git add . // add all changed files

```
git commit -m "message"
```

```
git push --set-upstream origin master  
// after this you can only do git push and it will push to above  
//specified branch
```

```
// to push to specific branch  
git push origin master
```

```
//to create new branch  
Git checkout -b <branch_name>
```

```
// BRANCH  
git branch // to view local branch  
git branch -r // to view remote branch
```

```
git checkout <branch_name> //to move from one branch to another
```

```
git merge < branch_name>  
// delete branch locally  
git branch -d localBranchName
```

```
// delete branch remotely  
git push origin --delete remoteBranchName
```

- stash

- NPM, Yarn

- Postman

HTML & CSS

[figma-design](#)

html

- syntax and css basic syntax.
 - <elements/tags attribute > </closing-element-tag>
- comments
- basics
 - head
 - meta
 - body
 - header
 - nav
 - main
 - section
 - article
 - aside
 - footer
- links
 - absolute vs **relative path**
- images
 - download
- list
- table
- form
 - input types
- html entities
- iframe

[most commonly use html tags](#)

[most commonly used html attributes](#)

CSS

- syntax
- selectors - differences
 - **combination of selectors**
 - **descendent selector** <space>
 - **direct child selector** >
 - **and selector**
 - **or selector**
 - **attribute selector**
 - **adjacent sibling selector** +
 - **general sibling selector** ~
- **specificity score**
- **box model**
 - **block vs inline**
- measurement units // % px em rem vh vw
- pseudo classes
 - **hover**
 - **focus**
 - visited
 - **active**
 - **first-child**
 - last-child
 - **nth-child**
 - only-child
 - of-first-type
 - **not(selector)** !
 - :selected
 -
- pseudo element
 - **before**
 - **after**
- **position**
- **flex**
 - direction :row (main-axis horizontal cross-axis vertical)
 - direction : column (main- axis vertical)
 - flex - container

- **direction** // when column (cross axis and main axis reverses too.)
 - **wrap**
 - **justify - content**
 - **align- items**
 - align - content
- flex - items
 - order
 - **flex-grow**
 - align-self
- **grid**
 - **grid-template-columns**
 - grid-template-rows
 - **grid-column**
 - **grid-row**
 - **gap**
- **media query**
 - desktop-first (max-with: <breakpoint>)
 - **mobile-first** (min-with : <breakpoint>)

BOOTSTRAP / Tawaidcss

[most commonly use css properties](#)

● JavaScript

- Variables (scope)
 - var (keyword)
 - Const (ES6 2015)
 - Let
- Naming
- Data Types
 - String
 - Number — integer, double, float
 - Boolean
 - null

- **undefined** // in js only
- **Object** {key:value}
 - **Array**
- .. **spread operator** // copies the value
- Functions
 - **Arrow function** =>
- Conditionals
- IF ELSE
 - **Ternary operator** ? :
- Operands and operators
 - assignment
 - arithmetic
 - comparison < <= > >= , == ,===
 - logical && || !
- Loops
- Array
 - **push**
 - unshift
 - **pop**
 - shift
 - splice
 - slice
 - **forEach**
 - **filter**
 - **Map** [1,2,3] => [2.4.6]
 - some => BOOLEAN
 - every => BOOLEAN
 - **find**
 - indexOf
 - findIndex
 - **includes**

String

- String functions
 - **replace()**
 - **replaceAll()**
 - **substring()**
 - slice()
 - **split()**
 - trim()
 - charAt()
 - toUpperCase()
 - toLowerCase()
- String properties
 - length
- Escape characters “\n” “\””

Template Literals

``randomstring${ variable }``

DOM manipulation

callback

Asynchronous

- **Promise / API**
- setTimeout

Call stack

Callback queue

Node JS:

[slides](#)

- Introduction to nodejs
- Node package manager (npm)
- * Global Object
- **Node modules** // import and export
- Node CLI
- Creating http server with nodejs
- Working on core NODE API
- Asynchronous and event loop in node js

Express JS:

[slides](#)

- Introduction to framework and Express
- The model-view-controller pattern in ExpressJS
- **Middlewares**
- **Routing**
- HTTP Protocol
- **Http Request Object**
- **Http Response Object**
- User **authentication using jwt** (json web token)
- Garbage collection and error handling

Mongo DB:

[slides](#)

- Installation
- Introduction SQL Vs NoSQL
- NoSQL (Schema Less Database)
- MongoDB (Collection & Documents)
- Data Modelling
- Create Database Shell Command
- Data Types
- Update, insert delete and query documents
- Using MongoDB Native driver with Node
- Database modeling using MongoDB
- **Aggregation** in mongodb
 - show dbs // view the existing databases
 - use <db_name> // use selected database
 - db // view the current database in use
 - show collections // view the list of collections
 - create collection // db.createCollection("<collection_name>")
 - db.dropDatabase()

db.<collection_name>.<method>

Create

- insertOne // {}
- insertMany // [{}, {}]

Read

- find()
- find({filterObject})
- find({filterObject}, {select})
- findOne({filterObject})

Update

- updateOne({filterObject}, {})
- updateMany({filterObject}, {})
- replaceOne({filterObject}, {})

Delete

- deleteOne({filterObject})
- deleteMany({filterObject})

Filter Operators // query operators

\$eq

\$ne

\$gt / \$gte

\$lt / \$lte

\$in

\$nin

\$not

\$and

\$or

\$exists

Update Operators

\$set

\$inc

\$rename

\$unset

Read Modifiers

sort

limit

skip

React JS / NEXT JS:

- Introduction Frontend tools (webpack , babel, more...) and React
- Single Page Application scope and objectives
- Dev Tool and Environment
- **JSX**, Babel, Webpack
- ES6 features practices
- Create-react-app and project setup
- **Components** patterns in React
- **State**

- **Props**
- **Life Cycle Dispatch**
 - **useSelector**
- **Error Handling**
 - **.catch**

lifecycle of React Components

- **Introduction to hooks**
- **React Router** // no need in nextjs
- **SPA using React Router**
- **props drilling**
- **State Management Redux**
 - **store**
 - **use**