

**MERN Stack Course Details**

**Kumari Galli – 2, Putalisadak**

**Kathmandu, Nepal**

**9847367755, 01-4221450**

**Copyright Reserved © Mind Risers Consortium Pvt. Ltd**

**MERN Stack Training Syllabus**

**install node**

**Introduction:**

• Introduction to JavaScript programming language

* Introduction to javascript
* Variables
  + - var (keyword)
    - **=Const** (ES6 2015)
    - **Let**
  + Naming
  + Data Types
  + Operands and operators
  + Conditionals
    - Ternary operator ? :
  + Functions
    - Arrow function
  + Loops
  + Array
* **push**
* unshift
* **pop**
* shift
* splice
* slice
* forEach
* **filter**
* **Map**
* some =>BOOLEAN
* every => BOOLEAN
* **find**
* **indexOf**
* findIndex
* **includes**

String

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

**Template Literals**

`randomstring${ variable } `

**callback**

**Asynchronous**

* Promise
* setTimeout

Call stack

Callback queue

• Technologies around JavaScript.

**Environment Setup:**

• Introduction to development tools

• Text Editor, IDE

• Git

• Git workflow(branching,push,pull)

• Git

Download from ( [download git](https://www.google.com/search?q=download+git&sxsrf=ALiCzsZQJ18VhXQeOV3vGQ5pNNL3m045yA%3A1651748550509&ei=xq5zYq7fHrzhz7sPpIiHsAY&ved=0ahUKEwiu2vHsmsj3AhW88HMBHSTEAWYQ4dUDCA4&uact=5&oq=download+git&gs_lcp=Cgdnd3Mtd2l6EAMyBAgAEEMyBAgAEEMyBAgAEEMyBQgAEIAEMgUIABCABDIFCAAQgAQyBQgAEIAEMgoIABCABBCHAhAUMgUIABCABDIFCAAQgAQ6BwgjELADECc6BwgAEEcQsAM6BwgAELADEEM6BwgjEOoCECc6BAgjECc6CggAELEDEIMBEEM6DQgAEIAEEIcCELEDEBQ6BQgAELEDOgsILhCABBCxAxCDAToHCAAQsQMQQzoLCAAQgAQQsQMQgwFKBAhBGABKBAhGGABQ9QRYtxlg1BxoAnABeAGAAbgBiAHfDpIBBDAuMTSYAQCgAQGwAQrIAQrAAQE&sclient=gws-wiz) )

“ 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 https://github.com/sagartmg2/express**

//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

GUI // git gui

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**

**MERN** [**slides**](https://docs.google.com/presentation/d/1JRxmIOM-6IiO-P0D63ksqgLqnd8WBRbY6IkEPD_dUdk/edit#slide=id.p)

**Node JS:**

[**slides**](https://docs.google.com/presentation/d/1roxJhAR1V7Q7S58CDvkil4UNWp5oFEJ7ynjDJd3nlaw/edit#slide=id.g12a4687e7c1_0_33)

• Introduction to nodejs

• Node package manager (npm

\* Global Object

• Node modules

• Node CLI

• Creating http server with nodejs

• Working on core NODE API

• Asynchronous and event loop in node js **Express JS:**

[**slides**](https://docs.google.com/presentation/d/1zA7R9WWPAXtAmWewVmt6YAYOAYYmYeWvjgGRCrNguPo/edit#slide=id.g12cd85df338_0_15)

• Introduction to framework and Express

• The model-view-controller pattern in ExpressJS

• Middlewares

• Templating engine (pug,handlebars)

• Routing

• HTTP Protocol

• Http Request Object

• Http Response Object

• User authentication using jwt (json web token)

• Garbage collection and error handling

**Mongo DB:**

[**slides**](https://docs.google.com/presentation/d/1rvmAyZQbewbqA3rXmuzakJmiFTNzNt2u/edit#slide=id.p1)

• Installation

• Introduction SQL Vs NoSQL

• NoSQL (Schema Less Database)

• MongoDB (Collection & Documents)

• Data Modelling

• Create Database Shell Command

• Datatypes

• Update, insert delete and query documents

• Using Mongodb Native driver with Node

• Database modeling using MongoDb

• Aggregation in mongodb

Filter Operators

$eq

$ne

$gt / $gte

$lt / $lte

$in

$nin

$and

$or

$not

$exists

Update Operators

$set

$inc

$rename

$unset

Read Modifiers

sort

limit

skip

**React 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

• Class, Inheritance, Encapsulation and more

• Create-react-app and project setup

• **Components** patterns in React

• **State**

• **Props**

• Life Cycle of React Components

• Functional Component

• Introduction to hooks

• React Router

• SPA using React Router

• Introduction to Redux and Flux Architecture

• State Management

• Reducers

• Action Creators

• Middlewares (thunk,saga)

• Services

• Error Handling