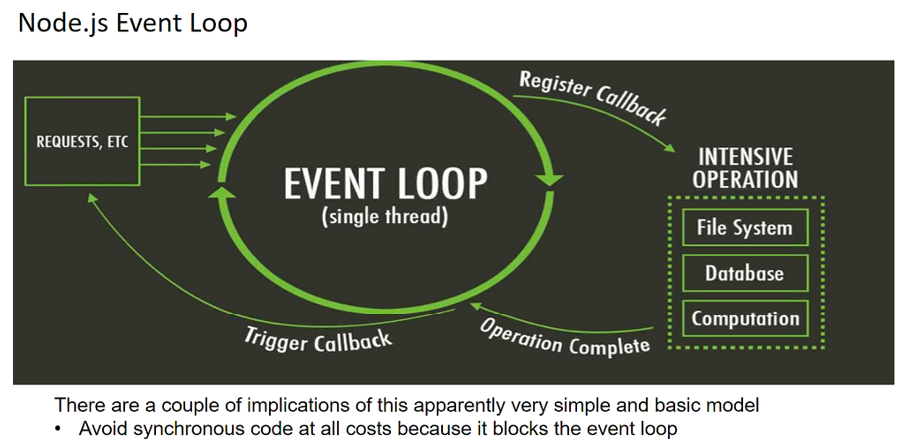
* Code Download - <https://1drv.ms/f/s!Ao-ceLq5rZm3hYdTjHIATb5WfxKQWA>
* ECMA Browser/Server Compatibility Matrix - Kagnax : <https://kangax.github.io/compat-table/es6/>
* Passport Authentication (**Authentication Middleware** for Node.js)- <http://passportjs.org/>
  + npm install passport --save
  + npm install passport-local --save
  + npm install express-session --save
  + npm install connect-flash --save
* IIFE – Immediate Invoke Function Expression
* Configuration Files - <https://github.com/lorenwest/node-config/wiki/Configuration-Files>
* Bower: <https://bower.io/>
* Package Manager for Web.
* Bower can manage components that contain HTML, CSS, JavaScript, fonts or even image files.
* Bower doesn’t concatenate or minify code or do anything else - it just installs the right versions of the packages you need and their dependencies.
* Bower keeps track of these packages in a manifest file, bower.json
* Bower is optimized for the front-end. If multiple packages depend on a package - jQuery for example - Bower will download jQuery just once. This is known as a **Flat Dependency Graph** and it helps reduce page load.
* npm install bower –g
* Created .bowerrc file to set Bower directory. (e.g. public/javascripts)
* bower init to create bower.json file
* Install Angular (Folders will be auto created under public/javascripts folder)
* bower install angular –save
* bower install angular-route –save
* JWT (JSON Web Token)
* JSON-based open standard for creating access tokens that assert some number of claims.
* For example, a server could generate a token that has the claim "logged in as admin" and provide that to a client.
* npm install jsonwebtoken --save
* Mongoose
* npm install mongoose --save
* Creating Schema
* Schema Types
* String
* Number
* Date
* Buffer
* Boolean
* Mixed
* ObjectId
* Array
* Built in Validators
* All Schema Types - required, unique
* Number - min, max
* String - enum, match, maxlength, minlength
* Methods available to use
* Read
* find({}) - Find All
* findOne({username:'abc'}) - Find One
* findById(1) - Find By ID
* Query
* find({}).where('')
* Update
* Find the Record
* Change the property Value
* Call Save()
* findOneAndUpdate({username:'abc'},{email:'xyz@test.com'})
* findByIdAndUpdate(1,{email:'xyz@test.com'})
* DELETE
* remove()
* findOneAndRemove({username:'abc'})
* findByIdAndRemove({username:'abc'})
* MEAN Stack:
* Mongo DB, Express JS, Node JS, Angular JS, (MEAN Stack)
* Server: (M E N)
* Client: (Angular / jQuery / JavaScript / React / KnockOut / Backbone)
* Web Frameworks
  + Express
  + Koa
* Learnjade.com
* Mongodb.com
* Robomongo (Client to use Mongo DB Server)
* Monk (Node js Mongo db driver)
* Mongoose (ORM Model)
* Browser/AppEngine/Runtime Grid

|  |  |  |
| --- | --- | --- |
| Browser | Layout Engine | Runtime |
| IE | Trident | Chakra |
| FF | Gecko | SpiderMonkey |
| Chrome | Webkit | V8 |
| Safari |  | Javascript Core |

* Language/Other Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| Language | C# | JAVA | JAVASCRIPT (MEAN Stack) |
| Core FW | .Net FW | JDK | Node.js |
| Runtime | CLR | JVM | Node.exe (v8 Engine) |
| Web FW | ASP .Net | JSP | Express JS (MVC) |
|  | ASP .NET MVC | SPRING |  |
|  | ASP .Net Web API | Java WebServices |  |
| Database | SQL Server | Oracle | Mongo DB |
| Client FW | Angular JS | Angular JS | Angular JS |

* Node.JS Event Loop:



* When to choose Node.JS
* Chat/Messaging
* Real Time Applications
* High Concurrent Aplications
* Communications Hubs
* Co-ordinators
* Intelligent Proxies
* Communication Hubs
* Use Node.JS For:
* Web Application
* WebSocket Server
* Ad Server
* Proxy Server
* Streaming Server
* Fast File Upload client
* Any Real Time Data Apps
* High I/O
* Database Matrix

|  |  |
| --- | --- |
| **RDBMS** | **MongoDB (No SQL)** |
| Database | Database |
| Table | Collection |
| Row / Tuple | Document |
| Column | Field |
| Join | Embedded Documents |
| Primary Key | Primary Key (\_id) |
| Oracle / SQL Server | Mongod |
| Sqlplus / SSMS | Mongo (RoboMongo) |