Main consumer oriented applicaition:

* Responsive behavior: This is browser dependent.
* Improvised User experience(UX):
* Cross browser:
* Cross Device: should have same experience irrespective of device.

For implementing above features this is concept came

* Progressive Web-Application: A progressive web app (PWA) is web app that uses modern web capabilities to deliver an app like experience to user.

The community came up with concept **modularity**

* PWA is guide lines to implement the modularity

Traditional WEB app:

We make HTTP request in HTML page (request goes to web hot server). Web server sends the HTML response.

Server sends the Java script (dynamic Java script). All browsers cannot perform same with the java script.

NEW TERMS: POST BACK, FRAME SET

The solution for above problem is **Single page APP**

**Single page app:**

When we make Http request for the first time it gives javaScript, which takes care of execution there on.

**Modern java script object**

**BROWSER:**

1. HTML Static DOM parser: The Static HTML rendering
2. Dynamic Java Script Parser JSOM:

* Object
* Events: On click etc
* Functions: functionality for on click event
* Types: datatypes

1. HTML 5 API system:

Different types like data type for date,time,month,range etc

* Drag drop API
* Worker: asynchronous
* Files
* Geolocation
* Sensors
* Canvas and SVG(Scalable Vector Graphics): pixel based graphics(media engine)
* Web worker:
* Media

1. Networking:

* Sockets
* Http (HTTP2 supports stream)

Browser Storages:

1. Session storage
2. Index DB
3. Local storage

**World of Java Scripts:**

* JQuery
* Java Script Libraries: e.g. JQuery (It works with only DOM), single responsbility

**Should be able to use multiple library (which makes front end complex)**

* That’s why came up with **Framework**

**Angular.js**

**Vue.js**

* UI: e.g. Bootstrap (to create data grid)
* Java Script Tools: for testing js (mostly libraries)

**ES 6 🡪 ES 2015, ES 7 🡪ES 2016, ES 8 🡪2017, ES Netxt**

**Standard for model JS programming:**

1. Classes
2. Generics
3. Arrow operators =>
4. Template string

* HTML interpolated string
* Parse string of HTML Object
* Simplification of String Concatenation

1. Scope using ‘let’ keyword
2. Modules using export and import

ES 6 🡪 Transpiler 🡪 ES 3

TypeScript, Modern JavaScript (High level JS / ES6) and Dart

Transpilers

* TypeScript 🡪 TSC
* ES 6 🡪 Bable, Tracure

**ISOMORPHIC Application:**

**MEAN (Mongo Angular Node)**

**Vue.js**

The progressive JavaScript Framework

* The core library is focused on the view layer, and is easy to pick up and integrate with other libraries or existing projects (works like library but structure is like framework).
* Vue is also perfectly capable of powering sophisticated Single page application (has routing)
* P framework can be inserted into you project you feel

Vue Instance:

Var obj = new vue({/\* <Properties> \*/})

Properties:

* el: Reference of DOM element for mounting the vue instance
* data: {}, JSON object, that contains properties/model to be loaded on DOM
* methods: {}, JSON
* computed:{},
* watch: {}, the properties those will watch other dependency properties
* Lifecycle service objects

Directives:

* Objects for

1. Binding

v-model directive, used to handle an event implicitely on element and update vue() instance