(3) Front End Technology

Page No.
Date

Web Framework :-

- Web framework is software framework which is designed to support the development of web application including web services, web resources & Web APIs.

- Web framwork is basically a software library that enables developers to write a software that runs on web.

- Web framework may be written in different language & using different methodologies.

- The term 'stack' is applied to refer to the collection of different language, software & framework.

* Features :- (1) Save Times
(2) Well Organized App
(3) Flexibility & Customizatable
(4) Code Reusability
(5) Security
(6)

* Popular Web framework

① Angular :- (Frontend Framework)
- Built by Google
- Well Supported
- Component-Based Architecture (UI built reusable Component)
- Two-Way Data Binding
- Powerful Command Line Interface :- Automater task like Component gen, testing & building Pro.

② React :- (Frontend Framework)
- Built by Facebook
- Component based Architecture
- Storage Ecosystem (Integrated with other libraries)
- Declarative UI
- Virtual DOM

③ Express (Backend framework)
- Uses Javascript
- Minimal & Fast
- Very customizable
- Very lightweight
- Support RESTful APIs.

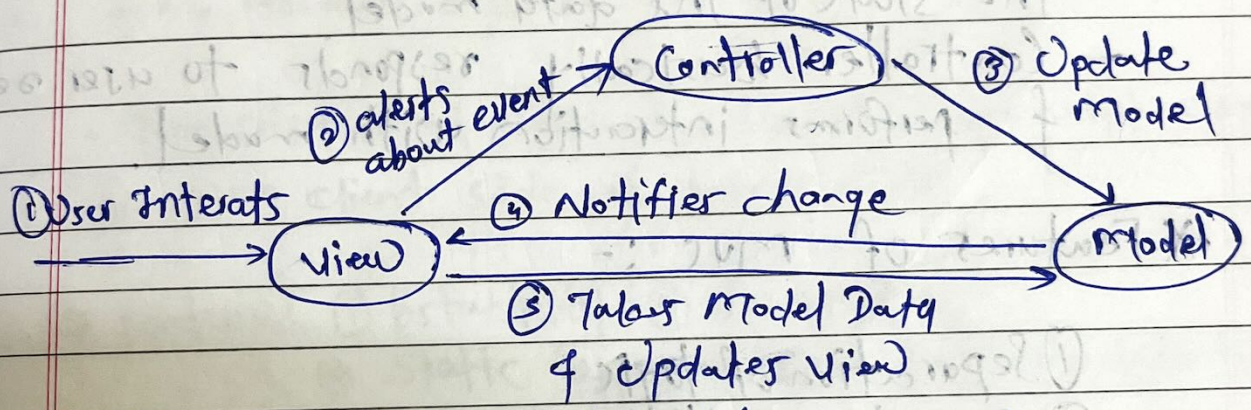
④ Bootstrap :-
- Built by Twitter
- Easy to learn
- looks Professional.

Model View Controller (MVC) :-

- The MVC stands for Model View Control.
- It is a pattern for the Architectural framework.

- MVC architecture is design pattern that divides an application into three main logical components (MVC)

- Each component handles a specific aspects ~~fo~~ of application.



- Model Represent the data
- View is the User Interface
- Controller is request Handler

① Model :-

- This part of architecture is responsible for managing application data.

- This module responds to the Request made from view.

- The model gives instruction to the controller to update when the response is made.

② View :-

- This part of architecture takes care of the presentation of data.
- The data is presented in desired format with the help of View
- This is script based system using JSP, ASP & PHP.

③ Controller :-

- The controller receiver input, validate it & then performs business operations that modify the state of the data model
- Controller basically responds to user request & performs interaction with model.

* Features of MVC :-

① Separation of logic
② Provide Multiple View
③ Faster Development Process
④ Customization
⑤ Returns data without formatting.

* TypeScript :-

- Typescript is superset of Javascript. Any valid javascript code is basically a valid for the typescript as well.

- Typescript extends Javascript by ~~addit~~ adding data types, classes & other object-oriented features

- Typescript is a programing language developed & maintained by microsoft.

- Typescript may used to develop Javascript appln for both client side & server side.

* Features ① Portability ② Javascript
③ Static Type checking ④ Support for library.

* Advantage :- ① It is an Open source language
② Portability (Runs on any browser)
③ Error detection (Identifies error at compile t)
④ Code Structuring
(Helps to organize & structure code)
⑤ Static Typing (allowing type to be added variable, function, properties)

* Disadvantage :- ① Does Not Support Abstract class
② It takes time to compile the code.
③ Learning Curve
④ Expressiveness Limitations.

Page No.
Date

* Sample Application in Typescript (Module)

- Module is used to set code written in Typescript. A module can be defined in a separate .ts file which can contain function, classe & interfaces.

- We use prefix <u>export</u> with all definations which we want to include the module & want to access from other modules

step ① = Hello-till create calculator.js module which contain class named Addition.

calculator.js =

```
export class Addition {
    a: number;
    b: number;
    constructor(a:number, b:number) {
        this.a = a;
        this.b = b; }
    display() {
        Console.log("Addition = "+(this.a+this.b));
    } }
```

step ② Now we will create a Javascript file for importing the class created above step.

app.ts

```
import {Addition} from './calculator';
let obj = new Addition(100,200)
obj.display();
```

Page No.
Date

* React JS :-

- React is OpenSource, Component-based front end JavaScript library maintained by facebook.

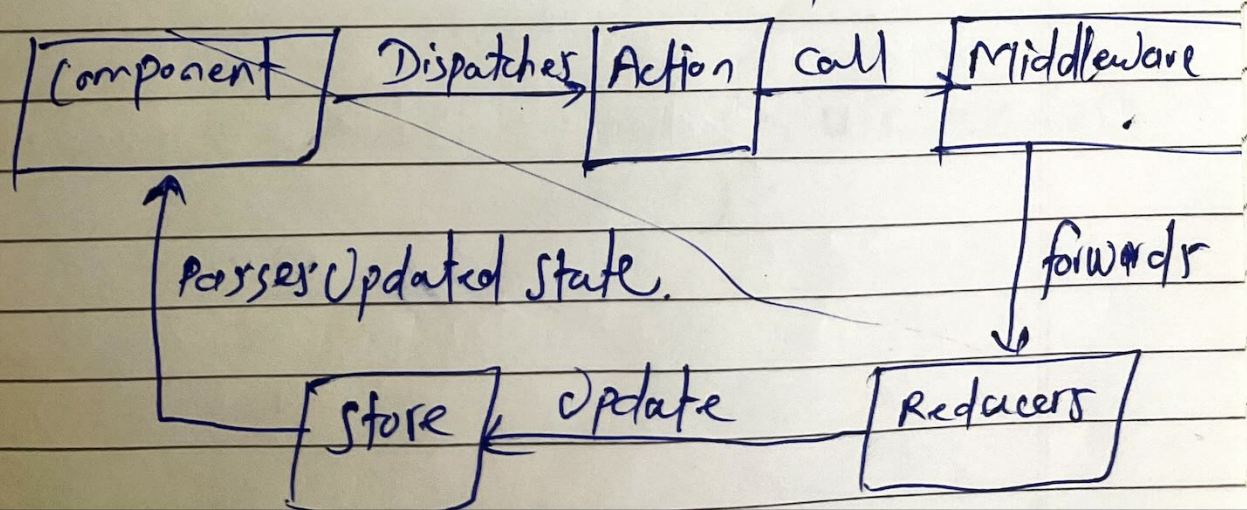
- This library is responsible only for view layer of the application. That means this JavaScript is for building user interface.

Features
1. Virtual DOM (Document Object Model)
2. Components based Architecture
3. One-Way Data Binding
4. Storage Ecosystem

* Redux :

- Redux is a Javascript library which can be used with ReactJS applications for the state management.

- Creating user interface React uses Redux. React Redux is official React building for Redux.



Page No.
Date

① Action :-
- Action is Redux represent work being done.
- It is simple javascript object which is used to send information from react component to reducer.

② Reducer :-
- Reducer is pure synchronous function which takes two things the previous state & action object.
- This function return the next state.

③ Store :- - It is centralized Place
- The store contain state of Application.
- Developers can access & update the store with the help of various methods.

④ Middleware :- It is used to execute the asynchronous code betn action & reducers.

Page No.
Date

* Hooks :-

- Hooks were added to React in version 16.8.
- Hooks allows the developers to use state & other React features without writing a class component.

- Hooks simplify code, make it readable and offer a more functional approach to React development.

* Rules of Hook :-

1. Only call the hooks at top level
2. Only call the hooks from React function
3. Hooks can not be conditional.

① UseState () :-

- UseState hook is used to declare state in functional Components
- It allow us to read & update the state with in the component.

Syntax :-

```
const [State, setState] = useState (initialState);
```

Example :-

```
const [count, setCount] = useState(0);
setCount (count + 1);
```

② useEffect() :-

- This hook allow you to perform side effects.
- These side effect can be fetching data, updating it, using timer & DOM manipulation.

Syntax :-

```
useEffect(<function>, <dependency>)
  //second parameter is optional
```

Example :-

```
useEffect(() => {
    console.log("Component Mounted");
  }, []);
```

③ useContext() hook :

- Context is technique which helps in passing data or state through component tree without having to pass props down manually.
- Purpose of using context is to share data which is considered as a global data.

Syntax :-

```
const contextvalue = useContext(MyContext);
```

Example :-

```
const user = UseContext(UserContext);
Console.log(user.name)
```

Page No.
Date

* Angular :-

- Angular is a Javascript framework which is used to create single page Application.

- The single page application is a kind of a application which dynamically rewrites the current page rather than loading entire new page from a server.

- Angular is completely based on component. Due to which application created in Angular gives the reactive speed.

- Angular appln are created with the help of HTML and Typescript.

Features :-

1. Support multiple Platform
2. High Speed Web Application
3. Productivity
4. Dynamic Development
5. Full Stack development.

Page No.
Date

* Pipe :-

- Pipes are used to transform the data
- That means piper take data as input & getting transformed into another format.

- The piper are written using pipe operator which is denoted by |.

- The pipe can be applied to any view or template & to any data input.

- Pipe in Angular is feauture used to transform data before displaying it in the view.

* Built in Pipes :

① date = Formats a date (May 20, 2025)
② uppercase = Converts text to Uppercase (HI)
③ lowercase = Converts text to lowercase (hi)
④ currency = Formats number as currency (₹1,00,000)
⑤ json = Converts object to JSON string {"name": "Ram"}

* Syntax :- {{ value | pipeName }}

Example :-

~~Main~~.ts :-

```
export class AppComponent {
    today : Date = new Date();
    name : String = 'angular';
    amount : number = 1000;
}
```

Page No.
Date

appComponent.html :-

```
<p> Todays Date : {{ today | date }} </p>
<p> Name Uppercase : {{ name | uppercase }} </p>
<p> Price : {{ amount | currency : 'INR' }} </p>
```

* Binding :-

① Property Binding :-

- It is one way binding which we can set the properties of element to the user interface page.
- It is used to bind from the Component (Typescript code) to an element's property in the HTML template.
- It helps in dynamically setting values of properties such as src, href, value, etc.
- It used square bracket [] around the HTML element property.

Example :-

AppComponent.ts :-

```
export class AppComponent {
    imgeUrl = " http://photo.jpg" ; }
```

AppComponent.html

```
<img [src] = " imgeUrl " >
```

[Model] → [View using property]

② Event binding :-

- It is used to listen to user events like clicks, key presses, mouse movements and call methods in the components when those event occurs.
- It uses parentheses () around the event name.

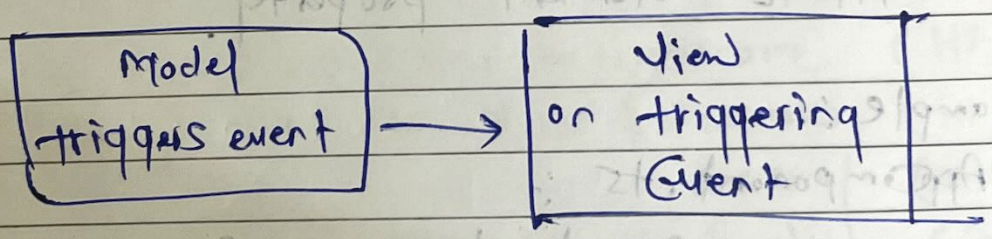
Example :-

AppComponent.ts :-

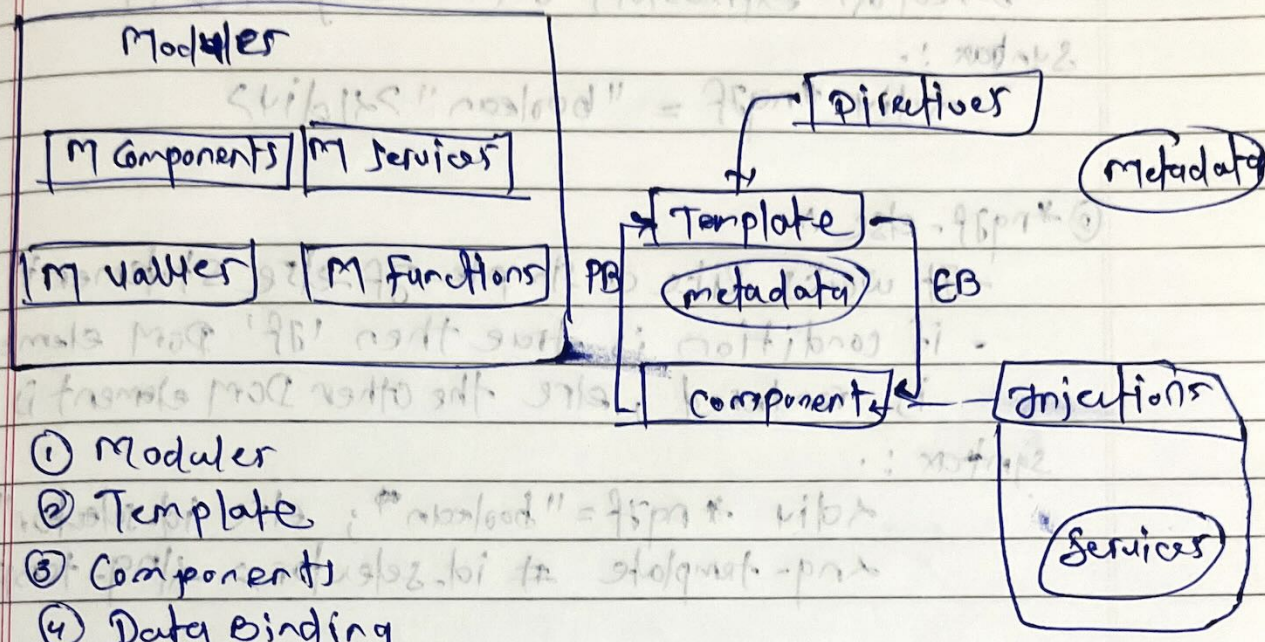
```
export class AppComponent {
    sayHello() {
        alert("Hello") ; }
```

App Component.html :-

```
<button (click) = "sayHello()"> ClickMe </button>
```



* Angular Architecture :-



1. Modules
2. Template
3. Components
4. Data Binding
5. Directives
6. Metadata
7. Injection of Services.

* Directives :-

- Structural directive are responsible for structure and layout of Dom Element.
- Structural Directive can be easily identified using the '*'. Every Structural directive is represented by a '*' symbol.

1. *ngIf
2. *ngIf-else
3. *ngFor
4. *ngSwitch.

① *ngIf :-

It is used to display or hide the Dom element based on expression value assigned to it.

Syntax :-

```
<div *ngIf = "boolean" ></div>
```

② *ngIf-else :-

- It works like a simple If-else statement,
- if condition is true then 'If' Dom element is rendered, else the other Dom element is rendered

syntax :-

```
<div *ngIf="boolean"; else id-selector"> </div>
<ng-template # id-selector> </ng-template>
```

③ *ngFor :-

- Is used to loop through the dynamic list in the Dom
- It is used to data presentation list & tables.

Syntax :-

```
<div *ngFor = "item of item-list"> </div>
```

④ *ngSwitch :-

It is used to choose between multiple case statement defined by expression inside the *ngSwitch & display on the Dom element.

Syntax :-

```
<div [ngSwitch] = "expression">
  <div *ngSwitchCase = "ex-1"></div>
  <div *ngSwitchCase = "ex-2"> </div>
</div>
```

⑤ Mobile Web Development

Page No.
Date

Mobile first :-

- Desktop, smartphone & tablet are having different screen size or layout to each other also user experience to website or application too.

- If the developer design website or web Appln for desktop layout or screen size then it is very difficult to design for mobile device such as smartphones, tablet etc.

- Designing & developing user interface for mobile before designing for desktop web or any other device is known as "Mobile first" Approach.

- Mobile first design is an approach of to web development where design & layout are optimized for mobile device first & then adapted for larger screen like tablets & desktops.

Mobile Web :-

- The use of internet can be through various way such as laptop, desktop, smartphones.

- This internet access is done with wireless or wired connection through these devices.

- As most of times users uses internet on their devices which is easy to accessible with in short period of time.

- Mobile device or smartphones are moveable so its flexible to access internet & web content.

- The use or access of internet and get web content through mobile device is known as a Mobile Web.

* Mobile Device :-
- Mobile device are also known as handheld computers or smartphones.
- Mobile device are portable that can connect to internet & designed for mobility.

Example :- ① Smartphone, ② Tablets
③ Smartwatches ④ eReaders
⑤ Foldable Phones, ⑥ Smartwatches.

* Desktop Device :-
- Desktop Device can not be move from one place to another place easily (Not Portable)
- They are large & heavy.
- These device are used by web designers & developers for better experience.

Example ① Computer ② Laptops
③ Mini PCs ④ All-in-One PCs.

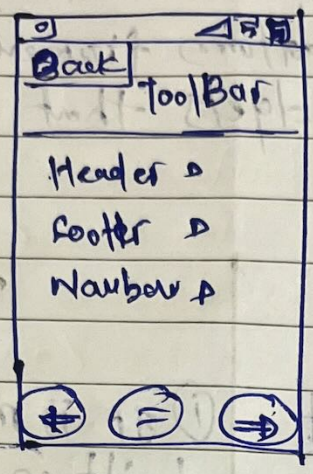
Page No.
Date

* JQuery Mobile :-

- JQuery Mobile also known as mobile framework which as open source, cross-platform framework. It provides touch friendly UI widgets that are designed for mobile devices.

JQuery	JQuery Mobile
① JQuery is fast, lightweight Javascript library used to simplify HTML Dom manipulation event handling & AJAX.	① It is mobile UI framework built on JQuery for creating responsive, touch-friendly mobile web apps.
② It has No-built-in UI widgets or themes.	② Includes ready-to-use UI components (button, lists, pages).
③ It has plain HTML + jQuery for interactivity	③ Uses HTML 5 data attributes like data-role = "Page"
④ It has a smaller size	④ It has larger size
⑤ Easier for beginners	⑤ Slightly more to learn due to data-role structure
⑥ It works on desktop & Mobile	⑥ Designed mainly for mobile & tablet device

- It has Powerful framework to style your mobile application



- You need to write your code once & it will work on Android, iphone, iPad or any other operating system. The same code can be run on web browser such as Google chrome, Safari, etc.

* Features : ① Accessibility
② Lightweight Size
③ Responsiveness
④ Theming & UI widgets
⑤ Simiplicity & Usability

* Advantage :- ① Easy to learn & Use
② Cross Platform Support
③ Saves Times
④ Built-in UI Components
⑤ Touch-friendly

* Disadvantage :- ① Limited functionality
② Application are slower on mobile device
③ Limited customization compared to modern framework
④ Low community activity

Page No.
Date

* Create A Page in JQuery mobile.

- The page is main unit of JQuery Mobile, the page is divided into three parts : header, footer and content.
- The main design part & manadatory section is content and will be declared using div tag

- The header divided into three sub areas - left, title and right where left & right area we can place a button & title we can set the title of the page
- The footer is optional.

```
<! DOCTYPE html>
<html>
<head>
<title> JQuery app </title>
<link rel = "stylesheet"
href = " http :// code.jquery.com/ ~~jquery~~-mobile/1.4.5/
        jquery.mobile-1.4.5.min.css
 />
<script src = " http ://code.jquery.com/jquery-1.11.1.min.js">
<script src = "http ://code.jquery.com/mobile/
        1.4.5/jquery.mobile-1.4.5.min.js >
</head>
<body>
 <div data-role = " page " >
 <div data-role = " header ">
  <h1> Page Header </h1>
  </div>
```

```
<div data-role="content">
  <h3> Page Main Content </h3>
  <p> ..... </p>
</div>
<div data-role="footer">
  <h4> Page footer </h4>
</div>
</div>
</body>
</html>
```

* Navigation (one page to another page)

```
<div page-role="content">
  <p> Main Content </p>
  <p> You can Go to the <a href="#page2">
    second page </p>
</div>
```

Exam

```
<div page-role="page" id="page2"
  data-title="This is second page"
    <div page-role="header">
      <h1> I am header </h1>
        <a href="#page2" data-icon="arrow-l"
          class="ui-btn-left"> Back </a>
    </div>
```

- Navigation is used for navigation from one page to another page.
- Create a navigation with btn click or a tag href property to new page.

Page No.
Date

* Layouts :-
- Layouts refers how the content is structured and display on a mobile screen.
- A JQuery Mobile layout is typically based on pages & each pages can have header, content & footer section.

① Single Page layout :-
- All Content is placed within a single div with data-role = "page".
- It is simplest form of layout.

```
<div data-rol = "page"> id = "home >
<div data-role = "header "> <h1> Home <1h1> <1div
        = "Content "> <p> Main content <1p>
        " Foter " > <h4) footer <1h4)
</div>
```

② Multi-Page layout :-
- A Single html file contain a multiple pages, each within its own div with data-role = "page"
- Navigation is smooth because page don't reload the whole document.

③ Fixed Toolbars layout :-
Headers and footer stay fixed at the top & bottom of the screen, even when scrolling.

```
"header" data-position = "fixied">
" Footer "  data-position = "fixed">
```

Page No.
Date

④ Grid layout :-

- It used to places the content in the columns.
- Grid system having classes ui-grid-a, ui-grid-b, ui-grid-c, ui-grid-d respectively.

```
<div class="ui-grid-a">
  <div class="ui-grid-a">
    <button> left </button>
  </div>
  <div class="ui-grid-b"> <button> Right
    </button> </div>
```

⑤ Dailog layout :-

- A popup-style page used for alerts, confirmation boxes or forms.

* Transitions :-

① slide
② slideup
③ slide Down
④ pop
⑤ Flip
⑥ Fade

Page No.
Date

* Widgets :-

- Widgets are pre-designed user interface (or) components that provide mobile friendly interaction & layouts.

① Button
② Listview
③ Navbar
④ Slider
⑤ Tops
⑥ Popup.

① Button Widget :-

- Transform normal HTML button, links or input element into touch-optimized styled button.
- Can include icon, themes and different button state (active, disable).

Example :-

```
<a href="#" data-role="button">
    click me </a>
```

② ListView Widget.

- It is used to create a scrollable list of item, such as menus or content listings.
- Support icons, dividers, split buttons.

```
<ul data-role="listview">
    <li> Home </li>
    <li> About </li>
    <li> Contact </li>
</ul>
```

③ Navbar Widget

- Creates a navigation bar usually placed at the top or bottom of the page
- Contain button or links for easy navigation

ex:

```
<div data-role="navbar">
<ul>
    <li><a href="#Home">Home</a></li>
    <li><a href="#Contact">Contact</a></li>
</ul>
</div>
```

④ Slidbar Widget:

- Convert an HTML range input into a touch-friendly slider
- Used for choosing numerical values like volume or brightness

Ex.

```
<label for="Volume">Volume</label>
<input type="range" id="Volume" "Volume"
        value="30" min="0" max="100">
```

* Events :-

① Mouse ~~click~~ Click :-

- It event happens when user presses & then releases a mouse button on an element.
- It is most common way users interact with buttons, links & other clickable elements on a webpage.

② Mouse Double Click :-

- The Mouse double click event occur when user quickly clicks the mouse button twice on the same element.
- It is different from single click & often used for special or advanced interactions.

③ Scroll Page :-

- The scroll event trigger when user moves the scrollbar on a webpage or a scrollable element.
- It detects vertical & horizontal movements within the content.

④ Page On Load :-

- The page load event fires when entire webpage & all of its resources (like image, scripts, etc) have fully loaded in the browser.

⑤ Keypress

⑥ Keydown.

```
<!DocTYPE html>
<html> <head>
 <title> JQuery App with Event </title>
 < link rel = " stylesheet "
 href = "http://jqcode.jquery.com/mobile/1.4.5/
                jquery.mobile-1.4.5.min.css "/>
 <script src = " http://code.jquery.com/jquery-1.11.1.
                min.js "> <script>
 <script src = " http://code.jquery.com/mobile/
                1.4.5/jquery.mobile-1.4.5.min.js"> </script>
 ~~head~~ <script>
 $(document).on ('pagecreate', '#Page1', function(){

① $("button").on ("click", function () {
        alert ("Button clicked ");
    });

② $ ("button ") .on ("dblclick", function c) {
        alert ( "Button Dabble-clicked );
    });

③ $ ("[data-role='content'] ").on ("scroll", function c){
        console.log ("Content being scrolled ");
    });

④ $ (document). on ('pageinit', '# Page1', function c){
        alert (" Page loaded !");
    });
 </script> </head>
 <body>
```

```
<div data-role="page" id="page1">
  <div data-role="header">
      <h1> First Page </h1>
  </div>
<div ~~content~~ data-role="content">
<button class="air-btn"> Click or Double Click Me </button>
  <P> ———— </P>
  <P> ———— </P>
  <P> ———— </P>
</div>
      <div data-role="footer">
        <h4> Publication </h4>
      </div>
    </div>
    </~~do~~body>
    </html>
```