jQuery Fast-Track Training Program (6 Days)

Audience: Trainees / Freshers
Duration: 5 Days
Objective: Learn jQuery concepts used in real-world and legacy projects

Day 1: jQuery Fundamentals, Syntax & Selectors

Topics Covered

1. Introduction to jQuery
2. History and use cases of jQuery
3. jQuery vs JavaScript
4. Advantages and limitations of jQuery
5. jQuery setup using CDN
6. jQuery syntax structure
7. `$` vs `jQuery`
8. jQuery object
9. `$(document).ready()`
10. jQuery comments
11. Chaining basics
12. `this` keyword
13. Debugging using browser console

Selectors

- ID selector
- Class selector
- Element selector
- Universal selector
- Group selector
- Attribute selectors
- Descendant and child selectors
- First, last, even, odd selectors

- `:eq()`, `:not()`, `:contains()`
- Form selectors (`:input`, `:text`, `:checked`)
- Visibility selectors (`:hidden`, `:visible`)

Implementation Scenario

- Interactive company website homepage
- Highlight active navigation menu
- Show welcome message after page load
- Handle basic user interactions

Day 2: Events, Effects, Animations & Styling

Event Handling

- Event binding and unbinding
- `click`, `dblclick`
- `mouseenter`, `mouseleave`
- `mouseover`, `mouseout`
- `hover`
- `focus`, `blur`
- `change`, `submit`
- Keyboard events (`keydown`, `keyup`, `keypress`)
- Scroll and resize events
- Event bubbling and delegation
- `on()` vs `off()`
- Prevent default behavior

Effects & Animations

- `show()`, `hide()`, `toggle()`
- `fadeIn()`, `fadeOut()`, `fadeToggle()`, `fadeTo()`
- `slideUp()`, `slideDown()`, `slideToggle()`
- Custom animations
- Animation queues

- Stop animations
- Callback functions

Styling

- `css()` single and multiple properties
- `addClass()`
- `removeClass()`
- `toggleClass()`
- `hasClass()`

Implementation Scenario

- FAQ accordion section
- Success and error notifications
- Mobile sidebar menu
- Light and dark theme switching

Day 3: DOM Manipulation & Form Handling

DOM Manipulation

- `text()`
- `html()`
- `val()`
- `attr()` and `removeAttr()`
- `prop()` and `removeProp()`
- `append()` and `prepend()`
- `before()` and `after()`
- `remove()`
- `empty()`
- Clone elements
- Wrap and unwrap elements

Form Handling

- Reading form values
- Checkbox handling
- Radio button handling
- Dropdown handling
- Multiple select fields
- File input basics
- Reset form using jQuery

Form Validation

- Required field validation
- Length validation
- Pattern validation
- Conditional validation
- Inline error messages
- Enable/disable submit button

Implementation Scenario

- To-Do / Task management system
- User registration form
- Dynamic survey form
- Preview form data before submission

Day 4: Traversing, Utilities & AJAX

DOM Traversing

- `parent()`, `parents()`, `parentsUntil()`
- `children()`
- `find()`
- `siblings()`
- `next()`, `nextAll()`, `nextUntil()`

- `prev()`, `prevAll()`, `prevUntil()`
- `closest()`
- Filtering elements

Utility Methods

- `each()`
- `map()`
- `data()`
- `$.extend()`
- `$.trim()`
- Type checking methods

AJAX

- Introduction to AJAX
- Asynchronous vs synchronous calls
- `load()`
- `$.get()`
- `$.post()`
- `$.ajax()`
- Sending and receiving data
- JSON handling
- Success, error, complete callbacks
- Loader and error handling

Implementation Scenario

- Load product list dynamically
- User profile dashboard
- Comments without page refresh
- API-based data display

Day 5 : Final jQuery Dynamic Table Demo

Demo Name

“Dynamic User Management Table”

Demo Definition

Build a **dynamic user table** on a single page with the following features:

- Add new rows dynamically (user info: Name, Email)
- Edit existing rows inline
- Delete rows
- Prevent duplicate names
- Show alert/notification messages for Add, Update, Delete, and Validation
- Use **parent-child DOM traversing** to locate elements
- All actions happen **without page reload**
- Minimal UI (table + input fields + buttons)

This demo demonstrates **selectors, events, DOM manipulation, traversing, validation, effects, and alerts** in a single, real-world scenario.

Required Functionalities

1. Add Row

- User enters Name and Email in input fields
- Clicking **“Add”** adds a new row in the table
- Validate:
  - Name and Email not empty
  - Name must be unique (no duplicates allowed)
- Show **success alert** on successful addition
- Highlight error alert if validation fails

jQuery Concepts Used:

- `val()`, `append()`, `parent()`, `children()`, `:contains()`
- Event handling: `click()`

- Validation logic

---

2. Edit Row

- Each row has an **Edit** button
- Clicking Edit:
  - Makes the Name/Email fields editable inline
  - Button changes to **Update**
- Clicking Update:
  - Updates the row with new data
  - Checks for duplicate Name
  - Shows **update alert**

jQuery Concepts Used:

- `text()`, `html()`, `siblings()`, `closest()`
- Traversing parent-child relationships
- Event delegation (`on()`) for dynamic rows
- Class manipulation to highlight editable fields

---

3. Delete Row

- Each row has a **Delete** button
- Clicking Delete:
  - Removes the row from the table
  - Shows **delete alert**

jQuery Concepts Used:

- `remove()`, `closest()`, `parent()`, `fadeOut()` (optional animation)

---

4. Alerts / Notifications

- Show **Add, Update, Delete, Validation error** messages
- Alerts disappear after 3 seconds
- Optional: Use fadeIn/fadeOut for animation

jQuery Concepts Used:

- Effects: `fadeIn()`, `fadeOut()`
- `setTimeout()`
- DOM manipulation for messages

---

5. Parent-Child Concept

- Table rows are **child elements** of `<tbody>` (parent)
- Buttons are **child elements** of each row
- Use `.closest('tr')` to find parent row from button click
- Use `.children('td')` to access cells

Example:

- Edit button → `.closest('tr')` → `.children('td')` → get Name/Email
- Delete button → `.closest('tr')` → `.remove()`

---

6. Optional Enhancements (Minimal Code)

- Disable Add button if fields are empty
- Highlight duplicate name error inline (optional)
- Alternate row coloring using `.addClass()` / `.removeClass()`

Day 6: Learn bellow Librarys and make proper demo

chart js – ref (https://www.highcharts.com/)
Jquery validation – ref (https://jqueryvalidation.org/)