**503-01 ADVANCE WEB DESIGNING**

**Assignment – 1**

**Q1) Explain CRUD in MongoDB with an example.**

**Answer:**  
**CRUD** stands for **Create, Read, Update, Delete** – the four basic operations of a database.

1. **Create (Insert Data):**

db.students.insertOne({ name: "Zack", age: 22, course: "BCA" });

1. **Read (Retrieve Data):**

db.students.find(); // Fetch all records

db.students.find({ name: "Zack" }); // Fetch specific record

1. **Update (Modify Data):**

db.students.updateOne({ name: "Zack" }, { $set: { age: 23 } });

1. **Delete (Remove Data):**

db.students.deleteOne({ name: "Zack" });

➡️ CRUD in MongoDB provides a simple way to manage documents in collections.

**Q2) What is DOM? Explain Directives in detail.**

**Answer:**  
**DOM (Document Object Model):**

* The DOM is a programming interface for HTML and XML documents.
* It represents the structure of a web page as a tree of objects, where elements (like <div>, <p>) can be accessed and modified with JavaScript.

**Directives in AngularJS:**  
Directives are markers on DOM elements (HTML tags) that tell AngularJS to attach specific behaviors or transform the DOM.

**Types of Directives:**

1. **ng-app** – Defines the root of an AngularJS application.
2. **ng-model** – Binds input fields with application data.
3. **ng-repeat** – Loops through a collection and repeats HTML elements.
4. **ng-if / ng-show / ng-hide** – Conditional rendering of elements.
5. **Custom Directives** – Developers can create their own directives for reusable components.

**Example:**

<div ng-app="">

<input type="text" ng-model="name">

<p>Hello, {{name}}</p>

</div>

➡️ DOM manages structure, while directives make it dynamic and interactive.

**Assignment – 2**

**Q1) Explain Event Handling in React.**

**Answer:**  
Event handling in React is similar to handling events in JavaScript but follows the React **synthetic event system** (cross-browser wrapper around native events).

**Steps for Event Handling in React:**

1. Define a function (event handler).
2. Attach it to an element using JSX syntax.

**Example:**

import React, { useState } from 'react';

function App() {

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

function handleClick() {

setCount(count + 1);

}

return (

<div>

<button onClick={handleClick}>Click Me</button>

<p>Count: {count}</p>

</div>

);

}

➡️ React uses camelCase (onClick, onChange) and functions are passed without parentheses.

**Q2) Explain expressions and filters in AngularJS.**

**Answer:**  
**Expressions in AngularJS:**

* Written inside {{ }} (double curly braces).
* Bind data from the model to the view.
* Can contain variables, operators, and functions.
* Example:

<p>Total: {{ price \* quantity }}</p>

**Filters in AngularJS:**

* Modify or format the value of an expression before displaying it.
* Common Filters:
  + **currency** – Formats as currency.
  + **date** – Formats date values.
  + **uppercase / lowercase** – Converts text.
  + **filter** – Filters an array based on condition.
* Example:

<p>{{ amount | currency }}</p>

<p>{{ name | uppercase }}</p>

➡️ Expressions show data, and filters refine how that data is displayed.

**Assignment – 3**

**Q1) Describe in detail how to validate forms in AngularJS.**

**Answer:**  
AngularJS provides built-in directives for form validation to ensure that user input is correct before submission.

**Steps for Form Validation:**

1. Use ng-model to bind form fields.
2. Apply validation directives such as required, ng-minlength, ng-pattern.
3. Use form states like $valid, $invalid, $pristine, $dirty.

**Example:**

<form name="myForm">

<input type="text" name="username" ng-model="username" required>

<span ng-show="myForm.username.$error.required">Username is required.</span>

</form>

➡️ AngularJS dynamically updates error messages and ensures data integrity.

**Q2) What is a module in AngularJS? Also explain how to create a controller of the module with an example.**

**Answer:**  
**Module:**

* A container for different parts of an AngularJS application (controllers, directives, filters, services).
* Defined using angular.module().

**Controller:**

* A JavaScript function associated with a module to control data and logic of the view.

**Example:**

var app = angular.module("myApp", []);

app.controller("myCtrl", function($scope) {

$scope.message = "Hello, AngularJS!";

});

<div ng-app="myApp" ng-controller="myCtrl">

<p>{{ message }}</p>

</div>

➡️ Modules organize the app, and controllers handle the data and functionality.

**Assignment – 4**

**Q1) Explain React Components in detail.**

**Answer:**  
**React Components** are the building blocks of a React application. They define how UI elements appear and behave.

**Types of Components:**

1. **Functional Components (Stateless):**
   * Simple JavaScript functions.
   * Use React Hooks (useState, useEffect).
   * Example:
2. function Welcome(props) {
3. return <h1>Hello, {props.name}</h1>;
4. }
5. **Class Components (Stateful):**
   * ES6 classes extending React.Component.
   * Can maintain state and lifecycle methods.
   * Example:
6. class Welcome extends React.Component {
7. render() {
8. return <h1>Hello, {this.props.name}</h1>;
9. }
10. }

**Features of Components:**

* Reusable and modular.
* Can have props (input data) and state (internal data).
* Follow a hierarchical structure.

➡️ React applications are basically a tree of nested components.

**Q2) Write about the features of ReactJS with its advantages and disadvantages.**

**Answer:**  
**Features of ReactJS:**

1. Virtual DOM for fast rendering.
2. Component-based architecture.
3. One-way data binding for predictable state management.
4. JSX (JavaScript XML) for writing HTML inside JS.
5. Supports server-side rendering.

**Advantages:**

* High performance due to Virtual DOM.
* Reusable components reduce development time.
* Large community and ecosystem.
* Works well with other libraries/frameworks.

**Disadvantages:**

* Learning curve for beginners.
* Frequent updates may cause compatibility issues.
* JSX syntax can be confusing at first.
* Requires additional libraries (like Redux) for complex state management.

➡️ ReactJS is a powerful library for modern web apps but requires proper structuring and additional tools for large projects.