**St. Francis Institute of Technology Borivli (West), Mumbai-400103**

**(Autonomous Institute)**

**Department of Information Technology**

**Academic Year: 2024-25**

**Class: TE-ITA/B                                                                            Semester: VI**

**Subject: Web Lab**

**Experiment – 5: To understand Events and Validations in AngularJS. (Create functions and add events, add HTML validators etc.)**

1. **Aim:** To study AngularJS functions and add events, add HTML validators
2. **Objectives:** Aim of this experiment is that, the students will be able

* To install AngularJS
* Read and understand commonly-used AngularJS syntax
* Environment setup

1. **Outcomes:** After study of this experiment, the students will be able

* To write code using AngularJS.
* to handle coding and syntax error

1. **Prerequisite:** Knowledge of HTML, CSS, JAVASCRIPT and any text editor.
2. **Requirements:** Personal Computer, Windows operating system, VSCode, Angular CLI, Node, browser, Internet Connection, google doc.
3. **Pre-Experiment Exercise:**

**Brief Theory:** Refer shared material

1. **Laboratory Exercise**
   * + 1. **Procedure:**

**a. Answer the following:**

### Explain $scope object in AngularJS

* Write about angular filters

**b**. **Attach screenshots:**

* AngularJS code and Output with your own comments and indentation.

1. **Post-Experiments Exercise**
2. **Extended Theory:**

Nil

1. **Questions:**

### What is the Angular Template variable, explain it with syntax and example ?

### explain AngularJS “required” and “email” client-side form validation.

## Two-Way Binding?

1. **Conclusion:**

* Write what was performed in the experiment.
* Write the significance of the topic studied in the experiment.

1. **References:**
2. <https://riptutorial.com/ebook/angularjs>
3. Practical AngularJS by DinisCruz

9. **Laboratory Exercise**

**B. Procedure:**

**Answer the following:**

### **Explain $scope object in AngularJS**

**ANS:** In AngularJS, the $scope object is a special JavaScript object that serves as a communication bridge between the controller and the view. It holds data (variables) and behaviors (functions) that can be accessed and manipulated within the AngularJS application. The $scope object plays a crucial role in data binding, event propagation, and inheritance within the application.Acts as a Bridge: Connects the controller and the view.

Two-Way Data Binding: Syncs model (JS) and view (HTML).

Scope Hierarchy:

* $rootScope: Global scope, accessible everywhere.
* Child scopes inherit from parent scopes.

Event Handling:

* $scope.$emit(): Sends events upward.
* $scope.$broadcast(): Sends events downward.
* $scope.$on(): Listens for events.

Dependency Injection: Automatically injected into controllers.

Prototypal Inheritance: Child scopes inherit properties from parents.

Digest Cycle: Watches changes and updates the DOM.

Limitations:

1. Complex nested scopes.
2. Performance issues due to frequent $digest() cycles.
3. Replaced by components in Angular 2+.
4. **Write about angular filters**

**ANS:** Filters in AngularJS enhance the presentation of data without modifying the original model. They help in formatting, sorting, and filtering data dynamically within expressions, making views more readable and user-friendly. While built-in filters handle common use cases, custom filters allow developers to define specific transformations as needed. However, excessive use of filters inside expressions can impact performance, so it's recommended to apply them in controllers or services when dealing with large datasets.Filters modify data before displaying it in the view without altering the original value.

1. Filters are applied using the | (pipe) symbol in expressions and directives.
2. Common built-in filters include currency, date, uppercase, lowercase, filter, orderBy, and number.
3. The currency filter formats numbers into currency format based on locale settings.
4. The date filter formats date values into readable formats like MM/dd/yyyy or fullDate.
5. The uppercase and lowercase filters convert text to upper or lower case.
6. The filter filter selects items from an array based on conditions like string matching or custom logic.
7. The orderBy filter sorts an array based on a specific field in ascending or descending order.
8. Custom filters can be created using app.filter() to define specific formatting or transformation logic.
9. Filters can be used in controllers and services by calling $filter('filterName')(data, parameters).

**b. Attach screenshots:**

1. AngularJS code and Output with your own comments and indentation.

**CODE:** INDEX.HTML

| <!DOCTYPE html>  <html lang="en" ng-app="ExpenseApp">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Smart Expense Tracker</title>  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">  <link rel="stylesheet" href="styles.css">  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>  <script src="app.js"></script>  </head><body ng-controller="ExpenseController" class="container mt-4"> | <h2 class="text-center text-primary">💰 Smart Expense Tracker 💰</h2>  <!-- Expense Form -->  <form ng-submit="addExpense()" class="card p-3 shadow">  <div class="mb-3">  <label class="form-label">Expense Name:</label>  <input type="text" ng-model="newExpense.name" class="form-control" required>  </div>  <div class="mb-3">  <label class="form-label">Amount (₹):</label>  <input type="number" ng-model="newExpense.amount" class="form-control" required min="1"> |
| --- | --- |

| </div>  <div class="mb-3">  <label class="form-label">Category:</label>  <select ng-model="newExpense.category" class="form-control" required>  <option value="Food">Food</option>  <option value="Transport">Transport</option>  <option value="Shopping">Shopping</option>  <option value="Other">Other</option>  </select>  </div>  <button type="submit" class="btn btn-primary w-100">➕ Add Expense</button>  </form>  <!-- Expense List -->  <div class="mt-4">  <h4 class="text-success">Expense List</h4>  <table class="table table-striped table-bordered">  <thead class="table-dark">  <tr>  <th>Name</th>  <th>Amount (₹)</th>  <th>Category</th> | <th>Action</th>  </tr>  </thead>  <tbody>  <tr ng-repeat="expense in expenses">  <td>{{ expense.name }}</td>  <td>₹{{ expense.amount }}</td>  <td>{{ expense.category }}</td>  <td>  <button class="btn btn-danger btn-sm" ng-click="removeExpense($index)">❌ Remove</button>  </td>  </tr>  </tbody>  </table>  </div>  <!-- Total Expense -->  <h3 class="text-center total-expense mt-4">Total Expense: ₹{{ getTotalExpense() }}</h3>    </body>  </html> |
| --- | --- |

APPJS FILE:

| var app = angular.module('ExpenseApp', []);  app.controller('ExpenseController', function($scope) {  $scope.expenses = [];  // Add Expense  $scope.addExpense = function() {  if ($scope.newExpense && $scope.newExpense.name && $scope.newExpense.amount && $scope.newExpense.category) {  $scope.expenses.push({  name: $scope.newExpense.name,  amount: parseFloat($scope.newExpense.amount),  category: $scope.newExpense.category | // Remove Expense  $scope.removeExpense = function(index) {  });  $scope.newExpense = {}; // Reset Form  }  }; $scope.expenses.splice(index, 1);  };  // Calculate Total Expense  $scope.getTotalExpense = function() {  return $scope.expenses.reduce((total, expense) => total + expense.amount, 0);  };  }); |
| --- | --- |

**OUTPUT:**



