

Experiment No.9

Name: Shardul Vikram Dharmadhikari.

Class: TE Comp.

Roll No: 8016.

Title:

Create an application for Bill Payment Record using AngularJS.

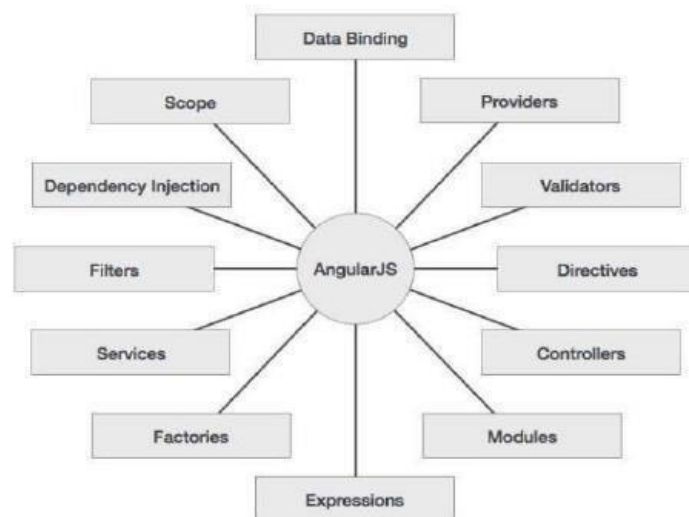
Theory:

AngularJS is an open-source web application framework. It was initially created in 2009 by Misko Hevery and Adam Abrons. It is presently kept up by Google. Its most recent adaptation is 1.2.21. "AngularJS is an auxiliary system for dynamic web applications. It gives you a chance to AngularJS is an open-source web application framework. It was initially created in 2009 by Misko Hevery and Adam Abrons. It is presently kept up by Google. Its most recent adaptation is 1.2.21. "AngularJS is an auxiliary system for dynamic web applications. It gives you a chance to utilize HTML as you layout dialect and gives you a chance to stretch out HTML's linguistic structure to express your application parts plainly and compactly. Its information official and reliance infusion take out a significant part of the code you as of now need to compose. Also, everything occurs inside the program, making it a perfect band together with any server innovation". **General Features:**

- AngularJS is a productive system that can make Rich Internet Applications (RIA).
- AngularJS gives designers a choices to compose customer side applications utilizing JavaScript in a spotless Model View Controller (MVC) way.
- Applications written in AngularJS are cross-program agreeable. AngularJS consequently handles JavaScript code reasonable for every program.

- AngularJS is open source, totally free, and utilized by a great many engineers the world over. It is authorized under the Apache permit version2.0.
- By and large, AngularJS is a system to assemble expansive scale, elite, and simple to-keep up web applications.

Core Features:



1. Data-authoritative: It is the programmed synchronization of information amongst model and view parts.
2. Scope: These are objects that allude to the model. They go about as paste amongst controller and view.
3. Controller: These are JavaScript capacities bound to a specific degree.
4. Services: AngularJS accompanies a few implicit administrations, for example, \$http to make aXMLHttpRequests. These are singleton objects which are instantiated just once in application.

5. Filters: These select a subset of things from a cluster and restore another exhibit.
6. Directives: Directives are markers on DOM components, for example, components, characteristics, css, and that's only the tip of the iceberg. These can be utilized to make custom HTML labels that fill in as new, custom gadgets. AngularJS has worked in mandates, for example, ngBind, ngModel, and so on.
7. Templates: These are the rendered see with data from the controller and model. These can be a solitary record, (for example, index.html) or different perspectives in a single page utilizing partials.
8. Routing: It is idea of exchanging sees.
9. Model View Whatever: MVW is an outline design for isolating an application into various parts called Model, View, and Controller, each with unmistakable obligations. AngularJS does not actualize MVC in the conventional sense, yet rather something nearer to MVVM (Model-View-ViewModel). The Angular JS group alludes it cleverly as Model View Whatever.
10. Deep Linking: Deep connecting permits to encode the condition of use in the URL with the goal that it can be bookmarked. The application would then be able to be re-established from the URL to a similar state.
11. Dependency Injection: AngularJS has a worked in reliance infusion subsystem that encourages the designer to make, comprehend, and test the applications effectively.
- 12.11. Dependency Injection: AngularJS has a worked in reliance infusion subsystem that
13. encourages the designer to make, comprehend, and test the applications effectively.

Advantages of AngularJS

- It gives the ability to make Single Page Application in a spotless and viable way.
- It gives information restricting ability to HTML. Along these lines, it gives client a rich and responsive experience.

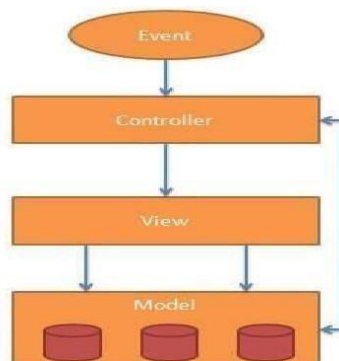
- AngularJS code is unit testable.
- AngularJS utilizations reliance infusion and make utilization of partition of concerns.
- AngularJS gives reusable segments.
- With AngularJS, the engineers can accomplish greater usefulness with short code.
- In AngularJS, sees are unadulterated html pages, and controllers written in JavaScript do the business handling.

Model View Controller

Model View Controller or MVC as it is famously called, is a product configuration design for creating web applications. A Model View Controller design is comprised of the accompanying three sections.

- Model – It is the most minimal level of the example in charge of looking after information.
- View – It is in charge of showing all or a part of the information to the client.
- Controller – It is a product Code that controls the connections between the Model and View.

MVC is mainstream since it secludes the application rationale from the UI layer and backings detachment of concerns. The controller gets all solicitations for the application and afterward works with the model to set up any information required by the view.



Model:

The model is in charge of overseeing application information. It reacts to the demand from see and to the directions from controller to refresh itself. The View:

An introduction of information in a specific arrangement, activated by the controller's choice to exhibit the information. They are content based layout frameworks, for example, JSP, ASP, PHP and simple to incorporate with AJAX innovation.

The Controller:

The controller reacts to client enter and performs communications on the information show objects. The controller gets input, approves it, and afterward performs business operations that alter the condition of the information demonstrate.

AngularJS is a MVC based structure

- An AngularJS application comprises of following three essential parts
–ng-app – This directive defines and links an AngularJS application to HTML.
- ng-model– This directive binds the values of AngularJS application data to HTML input controls.
- ng-bind – This directive binds the AngularJS Application data to HTML tags.

Program:

```
<html ng-app="billpayApp">
```

```
<!-- SCRIPTS TO BE ADDED IN HEAD TAG -->
```

```
<head>
```

```
<title>Bill Payment Record using angular and bootstram framework</title>
```

```
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
```

```
<!-- ACCESSING ANGULARJS BY CDN METHOD-->
```

```

<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.4/angular.min
.js"></script>

<!-- ACCESSING STYLESHEET FOR DESIGN [OPTIONAL PART CAN BE SKIP]-->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.mi
n.css">

<!-- MODEL PART-->

<script> var model = {
customer:
"Student", items: [{ bill:
"Electricity", status:
false
    },
    {
bill: "Internet(Wi/fi)", status:
false
    },
    {
bill: "Parking Charges",
status: false
    },
    {
bill: "Phone",
status: true
    },
    {
bill: "House Tax",
status: true
    }
  ]
}

varbillpayApp = angular.module("billpayApp", []); <html
ng-app="billpayApp">

```

```
<!-- SCRIPTS TO BE ADDED IN HEAD TAG -->
```

```
<head>
```

```
<title>Bill Payment Record using angular and bootstram framework</title>
```

```
<meta http-equiv="content-type" content="text/html; charset=utf-8" /> <!--  
ACCESSING ANGULARJS BY CDN METHOD-->
```

```
<script
```

```
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.4/angular.min.js"></s  
cript>
```

```
<!-- ACCESSING STYLESHEET FOR DESIGN [OPTIONAL PART CAN BE SKIP]-->
```

```
<link rel="stylesheet"
```

```
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.c  
ss">
```

```
<!-- MODEL PART-->
```

```
<script> var model
```

```
= {      customer:
```

```
"Student", items: [{ bill:
```

```
"Electricity", status:
```

```
false
```

```
    },
```

```
    {
```

```
bill: "Internet(Wi/fi)", status:
```

```
false
```

```
    },
```

```
    {
```

```
bill: "Parking Charges", status:
```

```
false
```

```
    },
```

```
    {
```

bill: "Phone", status:

true

},

{

bill: "House Tax", status:

true

}

]

}

varbillpayApp = angular.module("billpayApp", []);

billpayApp.controller("billpayctrl", function(\$scope) {

 \$scope.billpay = model;

 \$scope.dueBills = function() { var items

 = \$scope.billpay.items; var counter =

 0; items.forEach((item) =>

 { if (!item.status) { counter++;

 }

 })

 return counter;

 }

 \$scope.redFlag = function() {

 return \$scope.dueBills() <= 2 ? "label-success" : "label-danger";

 }

 \$scope.addBills = function(billName) { obj

 = { bill: billName, status: false

 }

 \$scope.billpay.items.push(obj);


```

    }
    $scope.removeBills = function(rmvBills) {
    $scope.billpay.items.splice($scope.billpay.items.indexOf(rmvBills), 1);
    }
    });
</script>
</head>
<!-- HTML BODY PART-->
<body ng-controller="billpayctrl">
<div class="container">
  <div class="page-header">
<h1>{{billpay.customer}}'s Bill's remained to Be Paid – <span class="lable"
ng-class="redFlag()" ng-hide="dueBills()==0">
    {{dueBills()}}
</span>
</h1>
</div>
<h3><center><b>Add extra biller fields if any</center></b></h3>
<div class="panel">
<div class="input-group">
<input class="form-control" ng-model="billName" /> <span
class="input-group-btn"> <button class="btnbtndanger" ng-
click="addBills(billName)">+ADD+</button>
</span>
</div>
<table class="table table-striped">
<thead>
<tr>

```

```
<th>Bill Name</th>
<th>Status</th>
<th>Status</th>
<th>Close</th>
</tr>
</thead>
<tbodyng-model="rmvBills">
<trng-repeat="item in billpay.items" ng-model="item">
<td>{{item.bill}}</td>
<td><input type="checkbox" ng-model="item.status" /></td>
<td>{{item.status}}</td>
<td>
<button type="button" ng-click="removeBills(item)">&times;</button>
</td>
</tr>
</tbody>
</table>
</div>
</div>
</div>
</body>
</html>
```

Output:



Student's Bill's remained to Be Paid - 3

Add extra biller fields if any

+ADD+

Bill Name	Status	Status	Close
Electricity	<input type="checkbox"/>	false	<input type="button" value="x"/>
Internet(Wifi)	<input type="checkbox"/>	false	<input type="button" value="x"/>
Parking Charges	<input type="checkbox"/>	false	<input type="button" value="x"/>
Phone	<input checked="" type="checkbox"/>	true	<input type="button" value="x"/>
House Tax	<input checked="" type="checkbox"/>	true	<input type="button" value="x"/>



Student's Bill's remained to Be Paid -

Add extra biller fields if any

+ADD+

Bill Name	Status	Status	Close
Internet(Wifi)	<input checked="" type="checkbox"/>	true	<input type="button" value="x"/>
Phone	<input checked="" type="checkbox"/>	true	<input type="button" value="x"/>
House Tax	<input checked="" type="checkbox"/>	true	<input type="button" value="x"/>



Conclusion: With the help of this assignment it is helpful to understand features of AngularJS. MVC model structure and its use in advanced web programming is studied.