Practical no: 9

Problem Statement: Create a login module for the web application using struts framework.

Name: Aditi Dinesh Mulay

Class: T.E. Computer

Subject: Web Technology

Div: A

Roll no: 02

PRN No. 71918146B

Figure.7: Architecture of AngularJS

Wt

Practical 9

Aditi Dinesh Nway
T.E. Comp Div: A
Roll no: 02

Title:- Design and Develop any web application using AngularJS.

Problem statement:- Create an application for Bill payment Record using AngularJS.

Outcomes: Student's can able to,
1) Implement the effective client side programming implentation.
2) Solve the complex problem of development using MVC framework.

Slw and Hlw Requirements: Eclipse IDE/ Notepad / Notepad ++, Web browser.

Theory: It is an open source web application framework.
- A proceduce productive system that can make Rich Internet Applications (RIA)
- Compose customer side applications utilizing Js in a MVC way.
- Applications written are cross-program ~~reg~~ readable AngularJs consequently handles Javascript code reasonable for every program.
- Open source, Totally free and utilized by a great many engineers the world over.
- By and large, AngularJS is a system to assemble expansive scale, elite and simple to keep up web applications.

Steps for AngularJS

1. When a link https://angularjs.org/ is opened, there are two options to download AngularJS **library –**

- **View on GitHub – Click on this button to go to GitHub and get all of the latest scripts.**
- **Download AngularJS 1 – Or click on this button, a screen as below would be seen –**

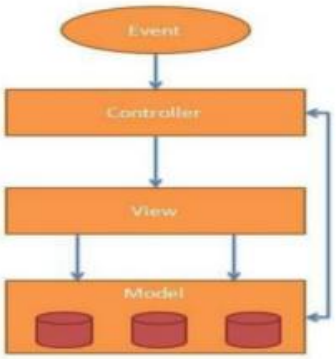


Figure. 8: Model View Controller

* Advantages

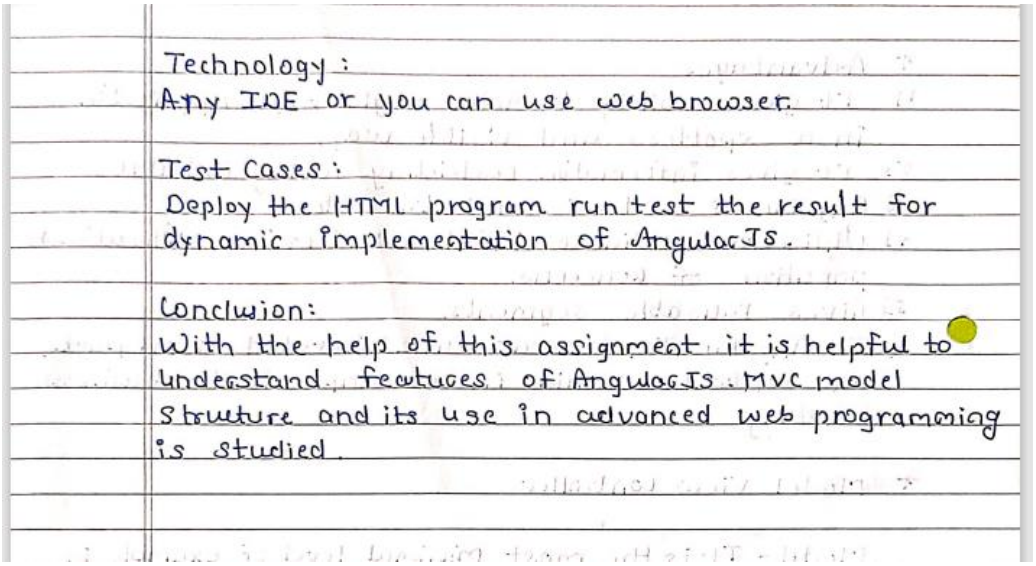
1) It gives ability to make single Page Application in a spotless and visible way.
2) It gives information restricting ability to HTML
3) Angular JS code is unit testable.
4) Utilizations reliance, infusion and make utilization of partition of Concerns.
5) Gives reusable segments.
6) In Angular JS, sees are unadulterated html pages, and controllers which in Javascript do the business handling.

* Model View Controller

Model - It is the most minimal level of 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.

* How Angular JS integrates with HTML.

→ ng-app directive indicates the start of AngularJS application.
→ ng-model directive then creates a model variable named "name" which can be used with html page.
→ ng-bind then uses the name model to be displayed in HTML span tag whenever use input something in text box.
→ Closing tag indicates the end of Angular JS application.



Example:

Now let us write a simple example using AngularJS library. Let us create an HTML file *myfirstexample.html* as below –

```
<!doctype html>
<html>

<head>
<scriptsrc="https://ajax.googleapis.com/ajax/libs/angularjs/1.5.2/angular.min.js"></script>
</head>

<bodyng-app="myapp">

<divng-controller="HelloController">
<h2>Welcome {{helloTo.title}} to the world of
Tutorialspoint!</h2> </div>

<script>
angular.module("myapp",[])
.controller("HelloController",function($scope){
        $scope.helloTo={};
        $scope.helloTo.title="AngularJS";
});
</script>
</body>
</html>
```

Following sections describe the above code in detail:

1. Include AngularJS

We have included the AngularJS JavaScript file in the HTML page so we can use AngularJS –

```
<head>
<scriptsrc="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
</head>
```

To update into latest version of Angular JS, use the following script source.

```
<head>
<scriptsrc="https://ajax.googleapis.com/ajax/libs/angularjs/1.5.2/angular.min.js"></script>
</head>
```

2. Point to AngularJS app

Next we tell what part of the HTML contains the AngularJS app. This done by adding the *ng-app* attribute to the root HTML element of the AngularJS app. You can either add it to *html* element or *body* element as shown **below –**

```
<bodyng-app="myapp">
</body>
```

3. View

The view is this part –

```
<divng-controller="HelloController">
<h2>Welcome {{helloTo.title}} to the world of Tutorialspoint!</h2>
</div>
```

ng-controller tells AngularJS what controller to use with this view. *helloTo.title*tells AngularJS to write the "model" value named helloTo.title to the HTML at this location.

4. Controller

The controller part is –

```
<script>
angular.module("myapp",[])
.controller("HelloController",function($scope){
    $scope.helloTo={};
    $scope.helloTo.title="AngularJS";
});
</script>
```

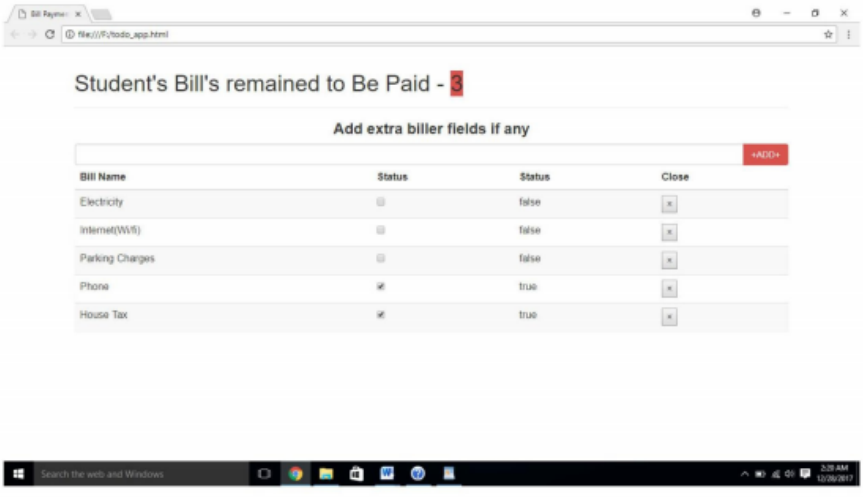
This code registers a controller function named *HelloController* in the angularmodule named *myapp*. The controller function is registered in angular via the angular.module(...).controller(...) function call.

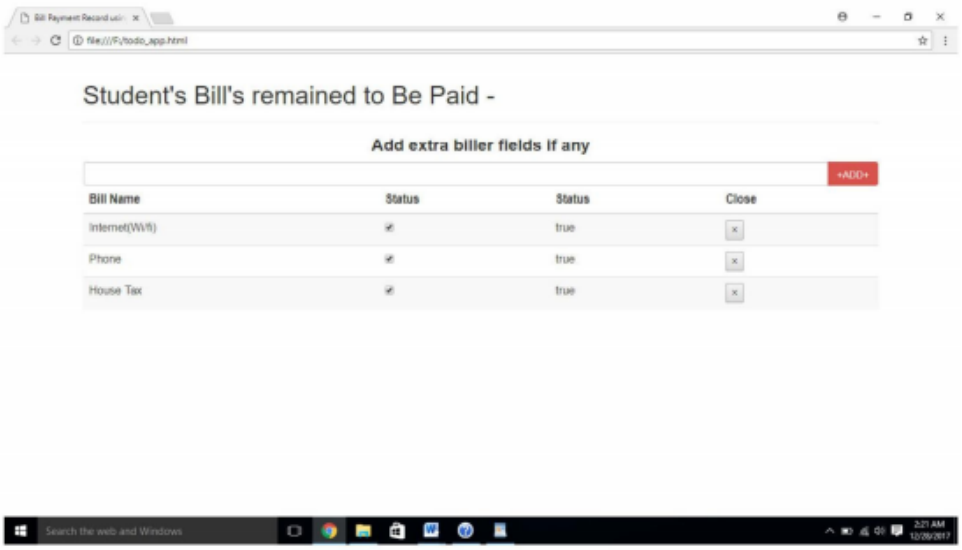
PROGRAM CODE: INPUT & OUTPUT

```
<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", []);
```

```
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="btnbtn-danger" 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>
```




Student's Bill's remained to Be Paid -
Add extra biller fields if any
+ADD+
Bill Name
Status
Status
Close
Internet(Wifi)
true
Phone
true
House Tax
true