

CLASS Assignment - 2

Q2.

Write a short note on .

a) Angular JS controller

→ They control the data of angular JS applications.

→ They are regular JS objects.

They controller in AngularJS is a javascript function that maintains the application data & behaviour using & Scope object.

Code :

```
<div ng-app="myAPP" ng-controller="myctrl">
  First name : <input type="text" ng-model="firstname" > <br>
  last name : <input type="text" ng-model="lastname" > <br>
  Full name : {{(firstname + " " + lastname)}}
</div>
```

```
<script>
var app = angular.module('myAPP', []);
app.controller("myctrl", function($scope) {
  $scope.firstname = "John";
  $scope.lastname = "Doe"
});
```

→ Angular JS application is defined by `ng-app="myApp"`.The application runs inside the `<div>`→ `"ng-controller = "myctrl"` is an AngularJS directive. It defines the controller→ `myctrl` is a javascript function.→ After invoking the controller with a `$scope` object, the controller creates two variables ("first name" & "last name")

The ng-model directives bind the input fields to the controller properties (First name & last name)

b) Angular JS scope: In angularJS, scope is a JS object that acts as a context for variables & functions available to the templates. It serves as the glue between the controller (JS) & the view (HTML).

→ The scope contains data that can be bound to the view, changes made to this data in the view or the controller are automatically reflected in both places due to angularJS's two-way data binding.

Code:

```
<div ng-app="myApp" ng-controller="myCtrl">
  <p> Name : {{name}} </p>
  <button ng-click="changeName()"> change Name </button>
</div>
<script>
  var app = angular.module("myApp", []);
  app.controller('myCtrl', function($scope) {
    $scope.name = "John Doe";
    $scope.changeName = function() {
      $scope.name = "Jane Doe";
    };
  });
</script>
```

Q4.

Ans

Explain any 5 angular JS directives with suitable example.

1) ng-app : This directive initializes on angular JS application. It designates the root element of the application & auto bootstraps it.

Code :

```
<div ng-app = "myApp" >  
</div>  
<Script>  
var app = angular.module('myApp', [ ]);  
</script>
```

2) ng-init : This directive initializes angularJS application data. It is often used to set initial values or execute expressions when the page loads.

Code :

```
<div ng-app = "myApp" ng-init = "name = 'John'">  
<p> Hello, {{name}} ! </p>  
</div>
```

3) ng-model : As mentioned earlier, this directive binds the values of HTML controls to the application data, enabling the two way data binding.

Code :

```
<div ng-app = "myApp" ng-controller = "myctrl">  
<input type = "text" ng-model = "name" />  
<p> Hello, {{name}} ! </p>  
</div>  
<Script>
```

" " // inc ng-model directives " "

```
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
  $scope.name = "John";
});
</script>
```

4) ng-repeat : This directive iterates over a collection & generates HTML for each item in the collection, allowing dynamic rendering of lists.

Code :

```
<ul>
<li ng-repeat="item in items">{{item}}</li>
</ul>
<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
  $scope.items = ['Apple', 'Banana', 'Orange'];
});
</script>
```

5) ng-if : This directive conditionally renders the HTML elements based on the evaluation of an expression. If the expression is truthy, the element is included in the DOM; otherwise it is removed.

(P.T.O)

Code :

```
<div ng-app = "myApp" ng-controller = "myCtrl">  
  <p > ng-if = "showElement" > This element is shown! </p>  
  </div>  
<script>  
var app = angular.module ('myApp', [ ] );  
app.controller ('myCtrl', function ($scope) {  
  $scope.showElement = true;  
});  
</script>
```

Q6. Write a flask code to implement GET & POST requests,

Ans from flask import Flask, request, json

```
app = Flask (__name__ )  
## GET request handler  
@app.route ('/ ', methods = ['GET'])  
def get_request_handler ():  
  return 'This is a get request'
```

Post request handler:

```
@app.route ('/ ', methods = ['POST'])  
def post_request_handler ():  
  data = request.get_json()
```

(P.T.O)

FOR EDUCATIONAL USE

→ It is collaborative by design.

FOR EDUCATIONAL USE

(P.T.O)

```

# Process the received data e.g: save to database
response-data = {'message': 'Data received successfully', 'data': data}
return jsonify(response-data)
if __name__ == '__main__':
    app.run(debug=True)

```

Q10.

Write a flask script to display "This is TSEC TE network".

Ans

```

from flask import Flask
# create a flask application
app = Flask(__name__)
# define a route for the root URL ('/')
@app.route('/')
def hello():
    return "This is TSEC TE Network"
if __name__ == '__main__':
    app.run(debug=True)

```

Q9.

Write short notes on DRUPAL, JOOMLA, DJANGO.

Ans

DRUPAL :

- It is a free, open source content management system to build & maintain websites, online directories, e-commerce stores, intranets & other types of digital content.
- It is a popular & customizable content management framework.

→ It includes thousands of modules & themes to attract web audiences & build a community.

Key Features : 1) It is known for "what you see is what you get", or WYSIWYG, content creation & editing tools. Even non-technical users can update & publish content.
2) It has the theme system to customize front-end.
3) It has drag-drop interface to build web-pages.
4) Enables users to add media to the website Other features :

- menu management
- URL management
- simple syndication
- log management
- content scheduling
- SEO

Layers in DRUPAL : Control flow works on 5 main layers.

1) Bottom (Base) layer is data layer.

2) Second layer → Modules.

3) Third layer → Blocks & menu's

4) Fourth layer → Permissions..

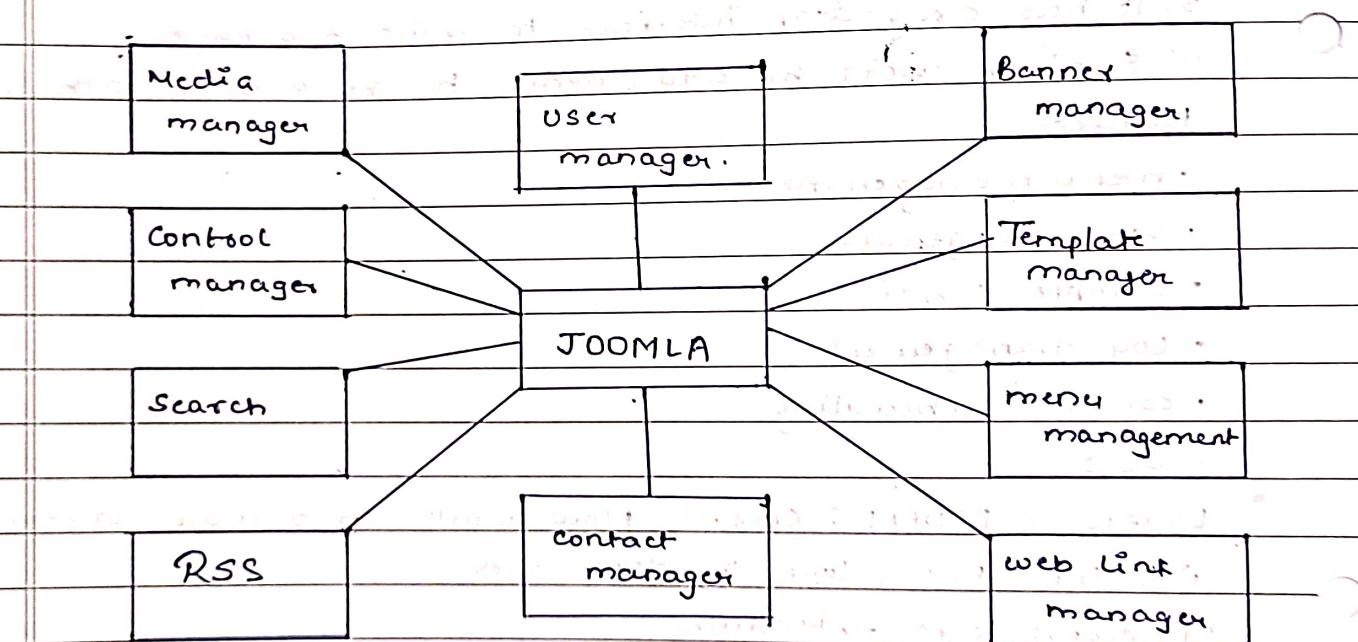
5) Fifth layer → The skin (HTML/CSS).

Advantages :

- It is open source & extensible.
- Highly customisable tools available.
- It includes web development facilities.
- Access the thousands of modules to get more user control.
- Easy to install & highly accessible.
- It is collaborative by design.

JOOMLA : It is an open source content management system which is used to build websites & online applications.

- It is free & extensible which is separated into front-end & back-end.
- It is developed using PHP, OOP. & MySQL.
- It is to be used by people who have basic website creation skills.



- user manager - It allows managing user information.
- content manager - It manages content using WYSIWYG.
- Banners manager - Used to add banners on the website.
- Template manager - Manages the design to be used on the website.
- Media manager - Easily uploading media.
- contact manager - Allows to add contracts & user information.

Advantages : 1) Open Source & Free

2) Easy to install & use.

3) Build website / app in less amount of time.

4) Very easy to edit content.

5) Data safety is present.

6) JOOMLA is compatible with all browsers.

Disadvantages : 1) Makes website heavy to load & run.

2) It is not SEO friendly.

3) Development is difficult to handle.

4) Plugins & modules are not free.

5) compatibility problems arise during installing.

DJANGO :

- It is a high level python web development framework that encourages rapid development & pragmatic design.
- It follows 'Don't repeat yourself (DRY)' principle aiming to minimize redundancy.
- It promotes code reusability.
- Provides built in features for handling common web development tasks . Like URL routing.
- It includes object-relational mapping (ORM) system for interacting with database . It makes working with data models easy .
- It has a vibrant community & extensive documentation making it easy for developers to find support .