Assignment-2

Q1) What is routing? Explain it with ng-route, ng-repeat, & ng-view.

→ Routing in AngularJS allows navigation between different views without reloading the page.
- This achieved using the ngRoute module.

Key Directives Related to Routing -
- ng-route: Enables in AngularJS by defining different views & controllers.
- ng-view: Acts as a placeholder for dynamically loaded views
- ng-repeat: Iterates over collections to dynamically generate UI elements.

Example -

```
<!DOCTYPE html>
<html>
<head>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular-route.js"></script>
</head>
<body ng-app="myApp">
    <a href="#!Home">Home</a> |
    <a href="#!about">About</a>
    <div ng-view></div>
    <script>
        var app = angular.module("myApp", ["ngRoute"]);
        app.config(function($routeProvider) {
            $routeProvider
            .when("/home", {template: "<h2>Welcome to Home</h2>"})
```

```
    .when("/about", {template: <h1> About us Page </h1>})
    .otherwise({redirectTo: "/home"});
});
</script>
</body>
</html>
```

Q2) Discuss controllers in AngularJS with examples.

→ A controller in AngularJS is a JavaScript function that manages application data & logic.

- It is defined using the ng-controller directive.

Example –

```
<!DOCTYPE html>
<html>
<head>
    <script src="https://ajax.googleapis.com/ajax/libs/
        angularjs/1.6.9/angular.min.js></script>
</head>
<body ng-app="myApp">
    <div ng-controller="myController">
        <h2> {{ message }} </h2>
    </div>
    <script>
        var app = angular.module("myApp", []);
        app.controller("myController", function($scope) {
            $scope.message = "Hello, AngularJS";
        });
    </script>
</body>
</html>
```

Here, the controller myController is defined inside myApp and the message is bound to view using {{message}}.

Qu3) Discuss default databases in MongoDB.
→ MongoDB has three default databases-
1) Admin:
- used for administrative purposes.
- Contains user roles, authentication data, and system-related command.

2) Local:
- stores temporary data & metadata.
- Does not replicate in a cluster.

3) Config:
- stores configuration settings in a shared ~~folder~~ cluster.
- Includes information about shards and chunk distribution.

These databases are essential for MongoDB's internal functionality.

Qu4) Explain Accessing & Manipulating Databases Commands in MongoDB.
→ MongoDB provides various commands to access & manipulate databases.

* Accessing databases -

use databasename

This command switches to databasename or creates it if doesn't exist.

* Manipulating Database Commands -

1) Create a Collection -

used to create a collection in database

db.createCollection("students").

2) Insert Data -

insertOne() → for single data

insertMany() → for multiple data.

eg

insertOne() -

db.students.insertOne({name:"John", age:22});

db.students.insertMany([{name:"Ram", age:21},
{name:"Shyam", age:23}]);

3) Find Data -

Used to search data in database

db.students.find().

4) Update Data -

Used to update the data in database

- updateOne()
- updateMany()

eg db.studentsupdateOne({name:"John"},{$set:{age:23}});

5) Delete Data -
Used to delete/remove the data.
- deleteOne()
- deleteMany()

eg
```
db.students.deleteOne({name: "Shyam"})
```

These commands helps in managing data efficiently in MongoDB.

Q45) How to handle cookies in python flask?
→ In flask, cookies are handled using the request & make_response modules.

*setting a cookie in flask -
```
from flask import Flask, request, make_response
app. = Flask(__name__)

@app.route('/setcookie')
def set_cookie():
    resp = make_response("cookie is set")
    resp.set_cookie("username", "JohnDoe", max_age=3600)
    return resp.
# Cookie expires in 1 hour.
```

*Getting a cookie in Flask.
```
@app.route('/getcookie')
def get_cookie():
    username = request.cookies.get("username")
```

return f"Welcome {username}"

* Deleting a cookie in Flask -

```
@app.route('/deletecookie')
def delete_cookie():
    resp = make_response("cookie deleted")
    resp.delete_cookie("username")
    return resp
```

This demonstrates how to set, & delete cookies in flask.

Q6) what is Flask URL building?

→ Flask URL building allows us to dynamically generates URL's for different routes using the url_for() function.

example -

```
from flask import Flask, url_for

app = Flask(__name__)

@app.route('/home')
def home():
    return "Welcome to home page!"

@app.route('/about')
def about():
    return "welcome to About Page"
```

```
@app.route('/get_urls')
def geturls():
    return f"Home URL {url_for('home')}
             About URL {url_for('about')}"
```

output -

If you visit /geturls, it returns

Home URL: /home
About URL: /about

This helps in maintaining clean & flexible URL's in flask applications.