Q.1) Create a Node.js file that demonstrate create Movie database and table in MySQL

(Slip 21)

(Using mongoose)

```
var mongoose=require('mongoose');

mongoose.connect('mongodb://0.0.0.0:27017/');

var db=mongoose.connection;

db.on('error',console.error.bind(console,'connection error'));

db.once('open',function()

{

console.log("Connection Successful!");


var movieSchema =mongoose.Schema(

   {

   title: String,

   director: String,

   year: Number,

   genre: [String]

});


var Movie = mongoose.model('Movie', movieSchema,'moviestore');


var newMovie = [{title: "Inception",director: "Christopher Nolan",year: 2010,genre: ["Sci-Fi",
"Action", "Thriller"]},

{title: "Rocketry",director: "ABC",year: 2023,genre: ["Sci-Fi", "Real incident"]}];


Movie.collection.insertMany(newMovie,function(err,docs)

{
   if(err)
      {
         return console.error(err);
      }
      else
      {
         console.log(newMovie);
      }
});
});
/*Output
Connection Successful!
[
  {
   title: 'Inception',
   director: 'Christopher Nolan',
   year: 2010,
   genre: [ 'Sci-Fi', 'Action', 'Thriller' ],
   _id: new ObjectId('6612a23f200ff604061976a2')
  },
  {
   title: 'Rocketry',
   director: 'ABC',
   year: 2023,
   genre: [ 'Sci-Fi', 'Real incident' ],
   _id: new ObjectId('6612a23f200ff604061976a3')
  }
]
*/
```

Q.2) Write node js application that transfer a file as an attachment on web and enables browser to prompt the user to download file using express js.

1)uploadFile.html(public folder)

```
<html>
  <head>
    <title>Sample</title>
    <style type="text/css">
      fieldset
      {
        width: 300px;
        margin-left: 550px;
        margin-top: 200px;
      }
    </style>
  </head>
  <body>
    <form action="uploadFile" method="post" enctype="multipart/form-data">
<!—action:link is given -->
      <fieldset >
        <legend align="center">Upload File</legend>
        <input type="file" id="filetoupload" name="filetoupload"><br><br>
       <input type="submit" value="upload">
      </fieldset>
    </form>
  </body>
</html>
```

2)DownloadFile.html (public folder)

```
<html>
  <head>
    <title>Download file</title>
    <style type="text/css">
      fieldset
      {
        width: 300px;
        margin-left: 550px;
        margin-top: 200px;
      }
    </style>
  </head>
  <body>
    <form action="downloadFile" method="post" enctype="multipart/form-data>
      <fieldset >
        <legend align="center">Download  File</legend>
<input type="submit" value="Download">
      </fieldset>
    </form>
  </body>
</html>
```

3)DownloadFile.js

```
const fs=require('fs');
var http=require('http');
var express=require('express');
var path=require('path');
var formidable=require('formidable');
var bodyparser=require('body-parser');
var app=express();
var newpath;
var server=http.createServer(app);
app.use( bodyparser.urlencoded( {extended:false} )  );
app.use(express.static(path.join(__dirname,'./public')))//html files saved in public directory
app.get('/',function(req,res)
{
   res.sendFile(path.join(__dirname,'./public/uploadFile.html'));
})

app.post('/uploadFile',function(req,res)//js file name
{
 var form=new formidable.IncomingForm();
 form.parse(req,function(err,fields,files)
 {
   var oldpath=files.filetoupload[0].filepath;

newpath='C:\Users\Administrator\Desktop\FG216'+files.filetoupload[0].originalFilename;//y
our folder path
   fs.rename(oldpath,newpath,function(err)
   {
       if (err) throw err;
      else
      {
       res.sendFile(path.join(__dirname,'./public/downloadFile.html'));
      }
   });
 });
});
```

```
app.post('/downloadFile',function(req,res)
{
   res.download(newpath,function(err)
   {
      res.end("Done with download...");
   });
});

server.listen(3030,function(){
   console.log("Server listening...");
})
```

Slip23

Q.1) Write node js script to interact with the file system, and serve a web page from a File

1)data.html

```
<html>
  <head>
    <title>Sample</title>
    <style type="text/css">
      fieldset
      {
        width: 300px;
        margin-left: 550px;
      }
    </style>
  </head>
  <body>
    <form>
      <fieldset >
        <legend align="center">Registration Form</legend>
        Student Name:<input type="text" placeholder="Enter your name"><br><br>
        Class:<select>
          <option value="Fymsc">FYMSC</option>
          <option value="Fymsc">SYMSC</option>
          <option value="Fymsc">TYMSC</option>
        </select><br><br>
        age:<input type="number" placeholder="Enter age"><br><br>
        address:<textarea rows="5" cols="20">here</textarea><br><br>
        <input type="submit" value="submit">&nbsp;
        <input type="reset" value="clear"
      </fieldset>
    </form>
  </body>
</html>
```

2)data.js

```
var http=require('http');
var fs=require('fs');
var data=fs.readFileSync('data.html');
http.createServer(function(req,res)
{
    res.writeHead(200,{ 'content-type': 'text/html' } );
    res.write(data);
    res.end();

}).listen(3030)
console.log("end");
```

Q.2) Write node js script to build Your Own Node.js Module. Use require ("http") module is a built in Node module that invokes the functionality of the HTTP library to create a local server. Also use the export statement to make functions in your module available externally. Create a new text file to contain the functions in your module called, "modules.js" and add this function to return today's date and time.

1)module.js

```
module.exports=function printDate()
{
    var d1=new Date();
    return (d1);
}
```

2)date.js

```
var http=require('http');
var req=require('./date_module.js');
d=req();

http.createServer(function(request,respose)
{
    respose.writeHead(200,{'content-type':'text/html'});
    respose.write(d.toString());
    respose.end();
}).listen(3030);


Output:
//Sun Apr 07 2024 16:18:25 GMT+0530 (India Standard
Time)...Shows on browser
```

Slip24

Q.2) Using angular js create a SPA to carry out validation for a username entered in a textbox. If the textbox is blank, alert “Enter username”. If the number of characters is less than three, alert ‟ Username is too short”. If value entered is appropriate the print “Valid username” and password should be minimum 8 characters

```
<!DOCTYPE html>
<html ng-app="validationApp">
<head>
  <title>Username Validation</title>
  <script src="angular.min.js"></script>
  <script>
    var validationApp = angular.module('validationApp', []);
 validationApp.controller('mainController', function ($scope)
      {
        $scope.validate = function ()
          {
          if (!$scope.username)
           {
             alert("Enter username");
          }
         else if ($scope.username.length < 3)
            {
             alert("Username is too short");
          }
          else
           {
             alert("Valid username");
          }
```

```
            if (!$scope.password)
              {
               alert("Password is required");
               }
            else if ($scope.password.length < 8)
             {
               alert("Password is too short");
             }
          };
        });
    </script>
</head>

<body ng-controller="mainController">
    <div>
        <h2>Username Validation</h2>
        <form>
            <label for="username">Username:</label>
            <input type="text" id="username" ng-model="username">
            <br><br>
            <label for="password">Password:</label>
            <input type="password" id="password"     ng-model="password">
            <br><br>
            <button type="button" ng-click="validate()">Validate</button>
        </form>
    </div>
</body>
</html>
```

Slip25:

Q.1) Create an angular JS Application that shows the location of the current web page. Slip25-1.html

```
<!DOCTYPE html>
<html ng-app="locationApp">
<head>
    <title>Current Page Location</title>
    <script src="angular.min.js"></script>
    <script>
        var locationApp = angular.module('locationApp', []);
 locationApp.controller('mainController', function ($scope,
$location)
      {
       $scope.currentLocation = $location.absUrl();
        });
   </script>
</head>
<body ng-controller="mainController">
    <div>
        <h2>Current Page Location</h2>
    <p>The current page location is: {{ currentLocation }}</p>
    </div>
</body>
</html>


<!--The current page location is:
file:///C:/Users/hp/OneDrive/Desktop/JS/Slip25-1.html#!/-->
```

Q.2) Create a js file named main.js for event-driven application. There should be a main loop that listens for events, and then triggers a callback function when one of those events is detected.(Slip25 ,22)

```
var events=require('events')
var em=new events.EventEmitter();

function add(a,b)
{
    console.log("Addition: "+(a+b));
}

function sub(a,b)
{
    console.log("subtraction: "+(a-b));
}

function mul(a,b)
{
    console.log("Multiplication: "+(a*b));
}

function div(a,b)
{
    console.log("Division: "+(a/b));
}

em.on('arithmetic',add);
em.on('arithmetic',mul);
em.on('arithmetic',sub);
em.on('arithmetic',div);

em.emit('arithmetic',12,2);

/*Addition: 14
Multiplication: 24
subtraction: 10
Division: 6*/
```

Slip22:

Q.1) Using node js create an Employee Registration Form validation.

EmployeeRegistration(folder)

-app.js

-data.js

-public(folder under EmployeeRegistration)

-index.html

-login.html

-registration.html

1)data.js

```
const userDB=[];
module.exports={userDB};
```

2)app.js

```
const express=require('express');
const http=require('http');
const bcrypt=require('bcrypt');
const path=require('path');
const bodyParser=require('body-parser');
const users=require('./data').userDB;

const app=express();
const server=http.createServer(app);

app.use(bodyParser.urlencoded({extended:false}));
app.use(express.static(path.join(__dirname,'./public')));
app.get('/',(req,res)=>{
    res.sendFile(path.join(__dirname,'./public/index.html'))
})
```

```
app.post('/register', async (req, res) => {
  try{
   let foundUser = users.find((data) => req.body.email ===
data.email);
  if (!foundUser)
  {
  let hashPassword = await bcrypt.hash(req.body.password, 10);

   newUser = {
                id: Date.now(),
                username: req.body.username,
                date:req.body.date,
                addr:req.body.addr,
                no:req.body.no,
                email: req.body.email,
                password: hashPassword,

            };
            users.push(newUser);
            console.log('User list', users);

 res.send("<div align ='center'><h2>Registration
successful</h2></div><br><br><div align='center'><a
href='./login.html'>login</a></div><br><br><div
align='center'><a href='./registration.html'>Register another
user</a></div>");
        }
else {
            res.send("<div align ='center'><h2>Email already
used</h2></div><br><br><div align='center'><a
href='./registration.html'>Register again</a></div>");
        }
```

```
    }
catch{
        res.send("Internal server error");
    }
});


app.post('/login', async (req, res) => {
    try{
      let foundUser = users.find((data) => req.body.email ===
data.email);
   if (foundUser)
      {
            let submittedPass = req.body.password;
            let storedPass = foundUser.password;


const passwordMatch = await bcrypt.compare(submittedPass,
storedPass);
     if (passwordMatch)
            {
                let usrname = foundUser.username;
 res.send(`<div align ='center'><h2>login
successful</h2></div><br><br><br><div align
='center'><h3>Hello ${usrname}</h3></div><br><br><div
align='center'><a href='./login.html'>logout</a></div>`);
            }
    else
          {
                res.send("<div align ='center'><h2>Invalid
password</h2></div><br><br><div align ='center'><a
href='./login.html'>login again</a></div>");
            }
        }
        else {
            res.send("<div align ='center'><h2>Invalid
email</h2></div><br><br><div align='center'><a
href='./login.html'>login again<a><div>");

        }

    }

 catch{

        res.send("Internal server error");

    }

});


server.listen(3000,function()

{

    console.log("server is listening on port :3000");

})
```

3)index.html

```
<html>

    <head>

      <meta charset="UTF-8">

      <title>My Form</title>

      <style>

        a{

            font-size: 40px;

        }

      </style>

    </head>

    <body align='center'>

        <form>

            <fieldset>

            <h1>Employee Registration</h1>
        <a href="./registration.html">Register</a>
        <br>
        <a href="./login.html">Login</a>
    </fieldset>
    </form>
    </body>
</html>
```

4)registration.html

```
<!DOCTYPE html>
<html lang = "en">
<head>
    <meta charset = "UTF-8">
    <title> My Form </title>
    <style>
        #mylink{
            font-size: 25px;
        }
    </style>
</head>
<body align='center'>
        <header>
        <h1>Employee Registration </h1>
    </header>

    <form action="/register" method="POST">
        <fieldset>
            <label>Employee Name</label>
            <input type ="text" id = 'username'
name="username" placeholder="maverick" required>
            <br><br>
            <label>Date Of birth</label>

            <input type ="date" id = 'date' name="date"
required>

            <br><br>


            <label>Address</label>

            <input type ="text" id = 'addr' name="addr"
placeholder="abc" required>

            <br><br>


            <label>Contact Number</label>

            <input type ="text" id = 'no' name="no"
placeholder="+91" required>

            <br><br>


            <label>Email ID</label>

            <input type ="email" id = 'email' name="email"
placeholder="abc@example.com" required>

            <br><br>


            <label>Password</label>

            <input type="password" id = "password"
name="password" required>

            <br><br>


            <button type ="reset">Reset</button>

            <button type ="submit">Submit</button>

        </fieldset>

    </form>

    <br><br>

        <a id="mylink" href="./login.html">login</a>
```

```
</body>
</html>
```

5)login.html

```
<!DOCTYPE html>
<html lang = "en">
<head>
    <meta charset = "UTF-8">
    <title> My Form </title>
    <style>
        #mylink{
            font-size: 25px;
        }
    </style>
</head>
<body align='center'>
        <header>
        <h1>Employee Login</h1>
    </header>

    <form action="/login" method="POST">
        <fieldset>

            <label>Email ID</label>
            <input type ="email" id = 'email' name="email"
placeholder="abc@example.com" required>
            <br><br>

            <label>Password</label>
            <input type="password" id = "password"
name="password" required>
            <br><br>


            <button type ="reset">Reset</button>
            <button type ="submit">Submit</button>
        </fieldset>
        </form>
        <br><br>
        <a id="mylink" href="./registration.html">register</a>


</body>
</html>
```