**EX.NO:09 MEAN STACK APPLICATION**

**DATE:**

**AIM:**

Creating a simple web application using MEAN STACK

**PROCEDURE:**

• Set up the components of MEAN stack- MongoDB, ExpressJS, AngularJS, NodeJS

• The json file is interesting because this file has all the dependencies and development-only dependencies of the project. Some MongoDB dependencies are added there later.

• Get the files for configuring the Typescript compilation.

• Get the json file that is for the Angular CLI.

• The e2e folder is for end-to-end testing and node\_modules stores all the dependencies that are listed in json.

• In the src folder, the angular application lives in. More config files and an app is obtained.

• In-app folder, the meet of our Angular application is written.

**CODE:**

Trial.html

<!DOCTYPE html>

<html>

<head>

<title>Register</title>

<style>

    body{

background-image: url(gymnn.jpg);

background-repeat: no-repeat;

 background-size: cover;

font-family: Arial, Helvetica, sans-serif;

font-size: large;

align-items: center;

text-align: center;

}

#block1{

    margin-left: 500px;

        margin-right: 400px;

        padding-right: 4px;

        background-color: white;

        border: 2px solid;

        border-color: white;

        border-radius: 10px;

        padding-left: 20px;

        padding-bottom: 100px;

        padding-top: 50px;

        font-family:'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;

        font-weight: bold;

        font-size: 17px;

        text-align:left;

}

button {

        background-color: #4caf50;

        width: 20%;

        color: orange;

        font-size: 20px;

        padding: 15px;

        margin: 10px 0px;

        border: none;

        cursor: pointer;

      }

      input[type="text"],

      input[type="password"]

      input[type="submit"] {

        width: 50%;

        margin: 8px 0;

        padding-top: 10px;

        display: inline-block;

        border: 1px solid green;

        box-sizing: border-box;

      }

      .ph {

        width: 30%;

        margin: 8px 0;

        padding-top: -8px;

        display: inline-block;

        border: 1px solid green;

        box-sizing: border-box;

      }

      #block2{

padding: 20px 100px;

}

  .fields{

    margin-left:-100px;

  }

  .links {

        background-color: rgb(43,35,35);

        border-radius: 0ex;

      }

      .link{

        color: white;

        margin-left: 1210px;

        margin-top:-20px;

      }

</style>

<script>

    function validateform() {

      var username = document.myform.username.value;

      var password = document.myform.password.value;

      var password2 = document.myform.password2.value;

      alert(username);

      if (username == null || username == "") {

        alert("UserName can't be blank");

        return false;

      }

      if (password.length < 7) {

        alert("Password must be at least 6 characters long.");

        return false;

      }

        if (password.length > 12) {

        alert("Password must not be greater than 12 characters");

        return false;

        }

        if(password != password2)

        {

          alert("Retype the same password");

        return false;

        }

    }

    function form() {

      alert("hello");

      var username=document.forms["myform"]["username"].value;

      //var username = document.myform.username.value;

      var password=document.forms["myform"]["password"].value;

      var password2=document.forms["myform"]["password2"].value;

      uid\_re=/^[A-Za-z0-9\_]{5,12}$/

      pwd\_re=/^(?=.\*[A-Z])(?=.\*[a-z])(?=.\*[0-9])(?=.\*[!@#$%^&\*])[A-Za-z0-9\_#@%\\*\-]{8,24}$/

      if (username == null || username == "") {

        alert("UserName can't be blank");

        return false;

      }

      if (password.length < 7) {

        alert("Password must be at least 6 characters long.");

        return false;

      }

        if (password.length > 12) {

        alert("Password must not be greater than 12 characters");

        return false;

        }

        if(password != password2)

        {

          alert("Retype the same password");

        return false;

        }

        if(!(uid\_re.test(document.forms["myform"]["username"].value)))

        {

        alert("Please enter a valid user ID with letters, numbers and underscore and with length between 5 and 12");

        return false;

        }

        if (!(pwd\_re.test(document.rform.pwd.value)))

      alert("Please enter a valid password with atleast one capital letter, small letter , a number and special character with length between 7 and 12");

         }

  </script>

  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js" >

  </script>

</head>

<body>

  <div class="links">

    <br />

    <a

      href="location.html"

      style="margin-left: 2%; margin-top: -3%; color: rgb(193, 216, 66)"

      ;

    >

      LOCATIONS ACROSS TAMILNADU

    </a>

    &nbsp; &nbsp;&nbsp; &nbsp;&nbsp; &nbsp;&nbsp; &nbsp;&nbsp; &nbsp;&nbsp;

    &nbsp;&nbsp; &nbsp;&nbsp; &nbsp;&nbsp; &nbsp;

    <a

      href="equp.html"

      style="margin-left: 2%; margin-top: -3%; color: rgb(193, 216, 66)"

      ;

    >

      LISTS OF EQUIPMENTS

    </a>

    &nbsp; &nbsp;&nbsp; &nbsp;&nbsp; &nbsp;&nbsp; &nbsp;&nbsp; &nbsp;&nbsp;

    &nbsp;&nbsp; &nbsp;&nbsp; &nbsp;&nbsp; &nbsp;

    <a

      href="gym\_homepage.html"

      style="margin-left: 2%; margin-top: -3%; color: rgb(193, 216, 66)"

      ;

    >

      HOME PAGE

    </a>

    &nbsp; &nbsp;&nbsp; &nbsp;&nbsp; &nbsp;&nbsp; &nbsp;&nbsp; &nbsp;&nbsp;

    &nbsp;&nbsp; &nbsp;&nbsp; &nbsp;&nbsp; &nbsp;

    <a

      href="gymcontentpage.html"

      style="margin-left: 2%; margin-top: -3%; color: rgb(193, 216, 66)"

      ;

    >

      ABOUT US

    </a>

    <br />

    <br />

  </div>

<div id="block1">

  <a href="cancel.html"  style="text-align:right;margin-left:230px">To cancel registered event or update phone</a>

<h3 style="background-color:black;color: floralwhite;text-align:center; ">Register for special events</h3><br>

<div id="block2" ng-app="">

<p>Please enter your details {{name}}!!</p>

<form id="myform" onsubmit="return form()" method ="post" action="http://localhost:3000/submit">

  <div class="fields">

<label for="name">Name:</label>

<input type="text" id="name" ng-model="name"

name="name" required><br>

<div class="boxes">

<label for="name">Username:</label>

<input type="text" id="username" name="username"

ng-model="username" required

onkeyup="move(this,this.value)"> <br>

<label for="password">password:</label>

<input type="text" id="password" name="password"

ng-model="password" required

onkeyup="move(this,this.value)"><br>

</div>

<label for="password2">Retype password:</label>

<input type="text" id="password2" name="password2"

ng-model="password2"  required

onkeyup="move(this,this.value)"> <br>

<label for="password2">Phone number:</label>

<input type="text" id="phno" name="phno"

ng-model="phno" required

onkeyup="move(this,this.value)"> <br>

<br>

</div>

    <input type="submit" value="submit" style="position: relative">

    </form>

    </div>

    <h5 style="text-align:right;margin-left:50px;">Your registration will be only once payment done </h5>

  <a href="payment.html" style="text-align:right;margin-left:300px;">Click here to proceed payment</a>

    </body>

</html>

Indexx.js

var express = require('express');

var path = require('path');

var bodyParser = require('body-parser');

var mongodb = require('mongodb');

var dbConn = mongodb.MongoClient.connect('mongodb://localhost:27017/node-demo',{

useNewUrlParser: true,

useUnifiedTopology: true

});

var app = express();

app.use(bodyParser.urlencoded({ extended: false }));

app.use(express.static(path.resolve(\_\_dirname, 'public')));

app.set('view engine','pug');

app.post('/submit', function (req, res) {

var data=req.body;

var record ={"Name":data.name,"username":data.username,"password":data.password,"password2":data.password2,"phno":data.phno}

dbConn.then(function(db) {

var dbo=db.db("powerzone");

//delete req.body.\_id; // for safety reasons

dbo.collection('programs').insertOne(record);

});

res.send('Data received:\n' + JSON.stringify(record));

});

 app.get("/", (req, res) => {

    res.sendFile(\_\_dirname + "/trial.html");

});

app.get('/all',function(req,res){

    dbConn.then(function(db) {

        var dbo=db.db("powerzone");

        dbo.collection("programs").find({},

        {"name":1,"username":0,"password":0,"password2":0,"phno":0}).toArray(function(err,result) {

        if (err) throw err;

        console.log(result);

        //res.send('All records:\n\n' + JSON.stringify(result));

        res.render('alpug', {result:result});

        });

        //var reco=dbo.collection('creditcard').find({});

        //console.log(reco);

        });

        });

        app.post('/delete', function (req, res) {

            dbConn.then(function(db) {

            var dbo=db.db("powerzone");

            var data=req.body.delname;

            var myquery = { Name: data };

            dbo.collection("programs").deleteMany(myquery, function(err,obj) {

            if (err) throw err;

            console.log(req.body.delname + " document(s) deleted");

            res.send(req.body.delname + " document(s) deleted");

            //db.close();

            });

            });

            });

            app.post('/update', function (req, res) {

            dbConn.then(function(db) {

            var dbo=db.db("powerzone");

            var myquery = {Name:req.body.oldname};

            console.log(req.body.oldname)

            console.log(req.body.newname)

            var newvalues = {$set:{Name:req.body.newname}};

            dbo.collection("programs").updateOne(myquery, newvalues,function(err, obj) {

            if (err) throw err;

            else{

            console.log("1 document updated");

            res.send("1 document updated");

            }

            });

        });

    });

    app.listen(3000,function(err){

    if (err) console.log(err);

    console.log("Server listening on PORT",3000);

    });

Alpug.pug

html

body

table(style='text-align:center;width:50%', border='1')

tr

th Name

=" "

=" "

=" "

=" "

=" "

=" "

=" "

th Username

=" "

=" "

=" "

=" "

=" "

th Phno

=" "

=" "

=" "

=" "

=" "

br

each item,i in result

 tr

  td=item.name

    =" "

  td=item.username

    =" "

  td=item.phno

 br

**OUTPUT:**

**Graphical user interface, application

Description automatically generated**

**PASSWORD VALIDATION:**

Graphical user interface, application

Description automatically generated

Graphical user interface, application

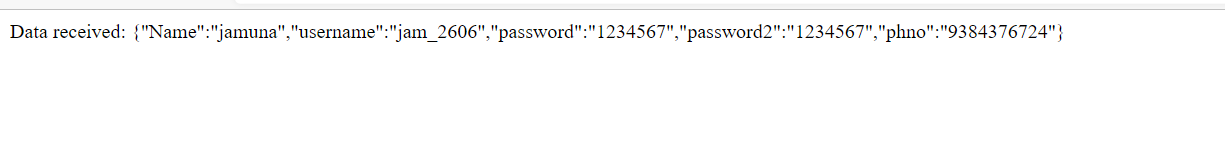
Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated



Database:

Graphical user interface, text, application, email

Description automatically generated

Cancellation:

A picture containing graphical user interface

Description automatically generated

Database:

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Creditcard.html

<!DOCTYPE html>

<html>

<head>

<title>Payment Gateway</title>

<meta charset="UTF-8">

<meta name="description" content="Payment Gateway">

<meta name="keywords" conent="Payment, Gateway, Debit Card, Credit

Card">

</head>

<style>

body{

background-image: url(loginimg.jpg);

background-repeat:no-repeat;

background-size:cover;

font-family: Arial, Helvetica, sans-serif;

font-size: large;

align-items: center;

text-align: center;

}

#block1{

background-color: white;

vertical-align: middle;

text-align:left;

width: 700px;

border: 15px #3a0d0d;

padding: 20px;

margin: 10px;

}

#block2{

padding: 5px 100px;

}

</style>

<body>

<script

src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"

>

</script>

<div id="block1">

<h3 style="background-color:black;color: floralwhite;text-align:center; ">Payment Details</h3><br>

<div id="block2" ng-app="">

<p>Please enter your details {{name}}!!</p>

<form id="payment" onsubmit="return validateForm()" method ="post" action="http://localhost:3000/submit">

<label for="name">Name:</label>

<input type="text" id="name" ng-model="name"

name="name"><br>

<div class="boxes">

<label for="cardno1">Card Number:</label>

<input type="text" id="cardno1" name="cardno1"

ng-model="cardno1" tabindex="1" class="card1" maxlength="4" size="4"

onkeyup="move(this,this.value)">

<input type="text" id="cardno2" name="cardno2"

ng-model="cardno2" tabindex="2" class="card2" maxlength="4" size="4"

onkeyup="move(this,this.value)">

<input type="text" id="cardno3" name="cardno3"

ng-model="cardno3" tabindex="3" class="card3" maxlength="4" size="4"

onkeyup="move(this,this.value)">

<input type="text" id="cardno4" name="cardno4"

ng-model="cardno4" tabindex="4" class="card4" maxlength="4" size="4"

onkeyup="move(this,this.value)">

<br>

</div>

<div class="form-group" id="exp-date">

    <label for="expdate">Expiry Date:</label>

    <select class="month" name="month\_id" id="month\_id"

    ng-model="month\_id">

    </select>

    <select class="year" name="year\_id" id="year\_id"

    ng-model="year\_id">

    <option value="2021"> 2021</option>

    </select>

    </div>

    <label for="cvv">CVV/CVC:</label>

    <input type="tel" name="cvv" id="cvv" ng-model="cvv"

    maxlength="3" size="3">

    <br><br>

    <img src="logos2.jpg" align="center" usemap="#workmap"

    width="180" height="60"><br><br>

    <input type="submit" value="Submit" style="position:

    relative">

    </form>

    </div>

    <form method="post" action="http://localhost:3000/delete">

        <label for="delname">Enter Name to delete:</label>

        <input type="text" id="delname" ng-model="delname"

        name="delname"><br>

        <input type="submit" value="Delete">

        </form>

        <br>

        <form method="post" action="http://localhost:3000/update">

        <label for="oldname">Old Name:</label>

        <input type="text" id="oldname" ng-model="oldname"

        name="oldname"><br>

        <label for="newname">New Name:</label>

        <input type="text" id="newname" ng-model="newname"

        name="newname"><br>

        <input type="submit" value="Update">

        </form>

        <br>

        <form method="get" action="http://localhost:3000/all">

        <input type="submit" value="View all Transactions">

        </form>

    </body>

    <script >

    //Using DOM to add months and years

    var yr=2021;

    var year = document.getElementById("year\_id");

    var month = document.querySelector(".month");

    for(let i=1;i<=12;i++)

    {

    let doc= document.createElement("option");

    doc.text=i;

    month.add(doc);

    }

    for(let i=1;i<=5;i++)

{

let doc= document.createElement("option");

doc.text=yr+i;

year.add(doc);

}

//Validating

function validateForm()

{

var cardno1 = document.forms["payment"]["cardno1"].value;

var cardno2 = document.forms["payment"]["cardno2"].value;

var cardno3 = document.forms["payment"]["cardno3"].value;

var cardno4 = document.forms["payment"]["cardno4"].value;

var cardno= cardno1+cardno2+cardno3+cardno4;

var uname = document.forms["payment"]["name"].value;

var cvv = document.forms["payment"]["cvv"].value;

var yr\_chosen = document.forms["payment"]["year\_id"].value;

var flag=1;

if(uname=="" || cardno=="" || cvv=="" )

{

window.alert("None of the fields can be empty!!");return

false;

}

else

{

//var cardno\_test = new RegExp("^\\d{16}");

var cvv\_test = new RegExp("^\\d{3}");

var name\_test = new RegExp("^[a-zA-Z. ]\*$");

var masterCard = new

RegExp("^(5[1-5][0-9]{14}|2(22[1-9][0-9]{12}|2[3-9][0-9]{13}|[3-6][0-9]{14}|7[0-1][0-9]{13}|720[0-9]{12}))$");

var visa = new RegExp("^4[0-9]{12}(?:[0-9]{3})?$");

var rupay = new RegExp("^6[0-9]{15}$");

if (!name\_test.test(uname))

{

alert("Invalid Name Field");flag=0;

}

if (!masterCard.test(cardno) && !visa.test(cardno) &&

!rupay.test(cardno))

{

window.alert("Invalid Card No.");

flag=0;

if (!cvv\_test.test(cvv))

{

alert("Invalid CVV");flag=0;

}

//alert(yr\_chosen);

if(yr\_chosen=="2021")

{

var d = new Date();

var n = d.getMonth();n++;

var mon\_chosen =

document.forms["payment"]["month\_id"].value;

if(mon\_chosen<n)

{

alert("Invalid Expiry Date");flag=0;

}

}

if(flag==1)

{

alert("Successful Submission!!");

return true;

}

return false;

}

}

}

function move(x,y)

{

if(y.length==x.maxLength)

{

let next = x.tabIndex;

if(next==1)

{

var cardno1 = document.forms["payment"]["cardno1"].value;

var rupay\_4 = new RegExp("^6[0-9]{3}$");

var visa\_4 = new RegExp("^4[0-9]{3}$");

var master\_4 = new RegExp("^5[1-5][0-9]{2}");

if (!visa\_4.test(cardno1) && !rupay\_4.test(cardno1) &&

!master\_4.test(cardno1))

{

window.alert("Invalid Card No.!!");

}

else

{

console.log(document.querySelector(".boxes").childElementCount);

if(next<document.querySelector(".boxes").childElementCount)

{

document.querySelector(".card" +

(next+1)).focus();

}

}

}

else

{

console.log(document.querySelector(".boxes").childElementCount);

if(next<document.querySelector(".boxes").childElementCount)

{

document.querySelector(".card" + (next+1)).focus();

}

}

}

}

</script>

</html>

Cred.js

var express = require('express');

var path = require('path');

var bodyParser = require('body-parser');

var mongodb = require('mongodb');

var dbConn = mongodb.MongoClient.connect('mongodb://localhost:27017/node-demo',{

useNewUrlParser: true,

useUnifiedTopology: true

});

var app = express();

app.use(bodyParser.urlencoded({ extended: false }));

app.use(express.static(path.resolve(\_\_dirname, 'public')));

app.set('view engine','pug');

app.post('/submit', function (req, res) {

var data=req.body;

var cardno= data.cardno1+data.cardno2+data.cardno3+data.cardno4;

var record ={"Name":data.name,"CardNo":cardno,"Month":data.month\_id,"Year":data.year\_id,"CVV":data.cvv}

dbConn.then(function(db) {

var dbo=db.db("webprog");

//delete req.body.\_id; // for safety reasons

dbo.collection('creditcard').insertOne(record);

});

res.send('Data received:\n' + JSON.stringify(record));

});

 app.get("/", (req, res) => {

    res.sendFile(\_\_dirname + "/CreditCard.html");

});

app.get('/all',function(req,res){

    dbConn.then(function(db) {

        var dbo=db.db("webprog");

        dbo.collection("creditcard").find({},

        {"Name":1,"CardNo":0,"Month":0,"Year":0,"CVV":0}).toArray(function(err,result) {

        if (err) throw err;

        console.log(result);

        //res.send('All records:\n\n' + JSON.stringify(result));

        res.render('allpug', {result:result});

        });

        //var reco=dbo.collection('creditcard').find({});

        //console.log(reco);

        });

        });

        app.post('/delete', function (req, res) {

            dbConn.then(function(db) {

            var dbo=db.db("webprog");

            var data=req.body.delname;

            var myquery = { Name: data };

            dbo.collection("creditcard").deleteMany(myquery, function(err,obj) {

            if (err) throw err;

            console.log(req.body.delname + " document(s) deleted");

            res.send(req.body.delname + " document(s) deleted");

            //db.close();

            });

            });

            });

            app.post('/update', function (req, res) {

            dbConn.then(function(db) {

            var dbo=db.db("webprog");

            var myquery = {Name:req.body.oldname};

            var newvalues = {$set:{Name:req.body.newname}};

            dbo.collection("creditcard").updateOne(myquery, newvalues,

            function(err, obj) {

            if (err) throw err;

            else{

            console.log("1 document updated");

            res.send("1 document updated");

            }

            });

        });

    });

    app.listen(3000,function(err){

    if (err) console.log(err);

    console.log("Server listening on PORT",3000);

    });

Allpug.html

html

body

table(style='text-align:center;width:50%', border='1')

tr

th Name

th CardNo

th Month

th Year

th CVV

<br>

each item,i in result

 tr

  td=item.Name

    =" "

  td=item.CardNo

    =" "

  td=item.Month

    =" "

  td=item.Year

    =" "

  td=item.CVV

OUTPUT:

A picture containing graphical user interface

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**RESULT:**

Thus , mean stack application is done successfully.