

Node js and mongo db connectivity

```
var mongo=require("mongodb").MongoClient

var url="mongodb://127.0.0.1:27017/"

mongo.connect(url,(err,db)=>{

  if(err) throw err

  console.log("Mongo connected")

})
```

CRUD OPERATIONSS

```
const express=require("express")
const mongo=require("mongodb").MongoClient
const app=express()
var url="mongodb://127.0.0.1:27017/session"
mongo.connect(url, (err,db)=>{
  if (err) throw err
  var dbc=db.db("bike")
  dbc.createCollection("data", (err,dat)=>{
    if(err) throw err
    console.log("collection created")
  })

  app.get('/insert', (req,res)=>{
    let data=[{model_name:'fz-250',make:"yamaha" ,color: 'black',
price:200000},
    {model_name:'rn5',make:"yamaha" ,color: 'blue', price:210000},
    {model_name:'continental-gt',make:"royal enfield" ,color:'silver',
price:400000},
    {model_name:'cbr-250',make:"honda" ,color: 'red', price:175000},
    {model_name:'xpulse',make:"hero" ,color: 'white', price:150000}
  ];
  dbc.collection("data").insertMany(data, (err,res)=>{
    if(err) throw err
    console.log("Data inserted")
  })
})

  app.get('/getdata', (req,res)=>{
    let data={model_name:"rn5"}
    dbc.collection("data").findOne(data, (err,res)=>{
      if (err) throw err
      console.log(res.make)
    })
  })

  app.get('/update', (req,res)=>{
    let data={color:'silver'}
    let updates={$set:{color:'red white'}}
    dbc.collection("data").updateOne(data,updates, (err,res)=>{
```

```

        if(err) throw err
        console.log(JSON.stringify("Updates done"+res))
    })
})

app.get('/delete', (req, res)=>{
    let data={model_name:"xpulse"}
    dbc.collection("data").deleteOne(data, (err, res)=>{
        if(err) throw err
        console.log("Deleted")
    })
})

app.listen(3000, ()=>{
    console.log("Server running")
})

```

DATE AND TIME

```

<!DOCTYPE html>
<html>
<body>

<h1>Show a Date and Time Control</h1>

<form>
  <label for="birthdaytime">Birthday (date and time):</label>
  <input type="datetime-local" id="birthdaytime" name="birthdaytime">
  <input type="submit">
</form>

</body>
</html>

```

Form Validation

```

<!DOCTYPE html>
<html>
<head>
  <title>Registration Form</title>
  <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
  <div class="container">
    <h2>Registration Form</h2>
    <form id="registrationForm" onsubmit="return validateForm()">
      <label for="username">Username:</label>
      <input type="text" id="username" required>
      <br>
      <label for="email">Email:</label>

```

```

<input type="email" id="email" required>
<br>
<label for="address">Address:</label>
<textarea id="address" required></textarea>
<br>
<label for="phone">Phone Number:</label>
<input type="text" id="phone" required>
<br>
<label for="password">Password:</label>
<input type="password" id="password" required>
<br>
<label for="confirmPassword">Confirm Password:</label>
<input type="password" id="confirmPassword" required>
<br>
<input type="checkbox" id="terms" required>
<label for="terms">I agree to the terms and conditions</label>
<br>
<input type="submit" value="Register">
</form>
</div>

<script src="script.js"></script>
</body>
</html>

```

CSS

```

.container {
  width: 300px;
  margin: 0 auto;
  padding: 20px;
  border: 1px solid #ccc;
  border-radius: 5px;
  text-align: center;
}

h2 {
  margin-bottom: 20px;
}

label {
  display: block;
  margin-bottom: 10px;
  text-align: left;
}

input[type="text"],

```

```
input[type="email"],
input[type="password"] {
  width: 100%;
  padding: 5px;
  margin-bottom: 10px;
}
```

```
textarea {
  width: 100%;
  padding: 5px;
  margin-bottom: 10px;
}
```

```
input[type="checkbox"] {
  margin-bottom: 10px;
}
```

```
input[type="submit"] {
  width: 100%;
  padding: 10px;
  background-color: #4CAF50;
  color: white;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}
```

```
input[type="submit"]:hover {
  background-color: #45a049;
}
```

JAVASCRIPT

```
function validateForm() {
  // Get the form input values
  var username = document.getElementById("username").value;
  var email = document.getElementById("email").value;
  var address = document.getElementById("address").value;
  var phone = document.getElementById("phone").value;
  var password = document.getElementById("password").value;
  var confirmPassword = document.getElementById("confirmPassword").value;
  var terms = document.getElementById("terms").checked;

  // Check if the username field is empty
  if (username.trim() === "") {
    alert("Username field is required");
    return false; // Prevent form submission
  }
}
```

```
}
```

```
// Check if the email field is empty or not a valid email format
```

```
if (email.trim() === "") {
```

```
    alert("Email field is required");
```

```
    return false; // Prevent form submission
```

```
} else {
```

```
    var emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
```

```
    if (!email.match(emailRegex)) {
```

```
        alert("Invalid email format");
```

```
        return false; // Prevent form submission
```

```
    }
```

```
}
```

```
// Check if the address field is empty
```

```
if (address.trim() === "") {
```

```
    alert("Address field is required");
```

```
    return false; // Prevent form submission
```

```
}
```

```
// Check if the phone field is empty or not a valid phone number format
```

```
if (phone.trim() === "") {
```

```
    alert("Phone Number field is required");
```

```
    return false; // Prevent form submission
```

```
} else {
```

```
    var phoneRegex = /^\d{10}$/; // assuming a 10-digit phone number format
```

```
    if (!phone.match(phoneRegex)) {
```

```
        alert("Invalid phone number format");
```

```
        return false; // Prevent form submission
```

```
    }
```

```
}
```

```
// Check if the password field is empty
```

```
if (password.trim() === "") {
```

```
    alert("Password field is required");
```

```
    return false; // Prevent form submission
```

```
}
```

```
// Check if the confirm password field is empty
```

```
if (confirmPassword.trim() === "") {
```

```
    alert("Confirm Password field is required");
```

```
    return false; // Prevent form submission
```

```
}
```

```
// Check if the password and confirm password match
```

```
if (password !== confirmPassword) {
```

```
    alert("Passwords do not match");
```

```

    return false; // Prevent form submission
}

// Check if the terms checkbox is checked
if (!terms) {
    alert("Please agree to the terms and conditions");
    return false; // Prevent form submission
}

// If all validations pass, you can perform additional actions or submit the form
// For example:
// document.getElementById("registrationForm").submit();

return true; // Allow form submission

```

Frames

```

<!DOCTYPE html>
<html>

    <head>
        <title>HTML Frames</title>
    </head>

    <frameset cols = "25%,50%,25%">
        <frame name = "left" src = "https://www.tutorialspoint.com/html/top_frame.htm" />
        <frame name = "center" src = "https://www.tutorialspoint.com/html/main_frame.htm"
    />
        <frame name = "right" src = "https://www.tutorialspoint.com/html/bottom_frame.htm"
    />

    <noframes>
        <body>Your browser does not support frames.</body>
    </noframes>
</frameset>

</html>

```

ROWSPAN

```

<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {

```

```
border: 1px solid black;
border-collapse: collapse;
}
</style>
</head>
<body>
```

<h2>Cell that spans two rows</h2>

<p>To make a cell span more than one row, use the rowspan attribute.</p>

```
<table style="width:100%">
  <tr>
    <th>Name</th>
    <td>Jill</td>
  </tr>
  <tr>
    <th rowspan="2">Phone</th>
    <td>555-1234</td>
  </tr>
  <tr>
    <td>555-8745</td>
  </tr>
</table>
</body>
</html>
```

Colspan

```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
</style>
</head>
<body>
```

<h2>Cell that spans two columns</h2>

<p>To make a cell span more than one column, use the colspan attribute.</p>

```
<table style="width:100%">
  <tr>
    <th colspan="2">Name</th>
    <th>Age</th>
```

```
</tr>
<tr>
  <td>Jill</td>
  <td>Smith</td>
  <td>43</td>
</tr>
<tr>
  <td>Eve</td>
  <td>Jackson</td>
  <td>57</td>
</tr>
</table>
</body>
</html>
```

Unordered list

```
<!DOCTYPE html>
<html>
<body>

<h2>An unordered HTML list</h2>

<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>

</body>
</html>
```

Ordered List

```
<!DOCTYPE html>
<html>
<body>

<h2>An ordered HTML list</h2>

<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>

</body>
```



```
</html>
```

Description list

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>A Description List</h2>
```

```
<dl>
```

```
<dt>Coffee</dt>
```

```
<dd>- black hot drink</dd>
```

```
<dt>Milk</dt>
```

```
<dd>- white cold drink</dd>
```

```
</dl>
```

```
</body>
```

```
</html>
```

Nested list

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>A Nested List</h2>
```

```
<p>Lists can be nested (list inside list):</p>
```

```
<ul>
```

```
<li>Coffee</li>
```

```
<li>Tea
```

```
<ul>
```

```
<li>Black tea</li>
```

```
<li>Green tea</li>
```

```
</ul>
```

```
</li>
```

```
<li>Milk</li>
```

```
</ul>
```

```
</body>
```

```
</html>
```

Horizontal List

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333333;
}

li {
  float: left;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 16px;
  text-decoration: none;
}

li a:hover {
  background-color: #111111;
}
</style>
</head>
<body>

<h2>Navigation Menu</h2>
<p>In this example, we use CSS to style the list horizontally, to create a navigation
menu:</p>

<ul>
  <li><a href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>

</body>
</html>
```

Letters in ordered list

```
<!DOCTYPE html>
<html>
<body>

<h2>Ordered List with Letters</h2>

<ol type="A">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>

</body>
</html>
```

Geolocation

```
<!DOCTYPE html>
<html>
<body>

<p>Click the button to get your coordinates.</p>

<button onclick="getLocation()">Try It</button>

<p id="demo"></p>

<script>
var x = document.getElementById("demo");

function getLocation() {
  if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(showPosition);
  } else {
    x.innerHTML = "Geolocation is not supported by this browser.";
  }
}

function showPosition(position) {
  x.innerHTML = "Latitude: " + position.coords.latitude +
  "<br>Longitude: " + position.coords.longitude;
}
</script>

</body>
```

</html>

JavaScript Validation

create a javascript validation form that satisfy the following property

- 1.user name field not empty
- 2.password validation
 - a)lower case
 - b)upper case
 - c)special
 - d)number
 - e)minimum length
- 3.list box exclude few countries
- 4.date restriction
- 5.phone number
 - a)isnumber
 - b)length
- 6.email id
- 7.password confirmation

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Form Validation</title>
```

```
<script>
```

```
function validateForm() {
```

```
    // User name field not empty
```

```
    var username = document.forms["myForm"]["username"].value;
```

```
    if (username == "") {
```

```
        alert("Please enter your username.");
```

```
        return false;
```

```
    }
```

```
    // Password validation
```

```
    var password = document.forms["myForm"]["password"].value;
```

```
    var passwordRegex = /^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?=.*[!@#$%^&*]).{8,}$/;
```

```
    if (!password.match(passwordRegex)) {
```

```
        alert("Please enter a valid password. It must contain at least one lowercase  
letter, one uppercase letter, one special character, one number, and be at least 8  
characters long.");
```

```
        return false;
```

```
}
```

```
// List box excluding countries
```

```
var country = document.forms["myForm"]["country"].value;  
var excludedCountries = ["Country1", "Country2", "Country3"];  
if (excludedCountries.includes(country)) {  
    alert("Selected country is not allowed.");  
    return false;  
}
```

```
// Date restriction
```

```
var selectedDate = document.forms["myForm"]["date"].value;  
var currentDate = new Date();  
var selectedDateTime = new Date(selectedDate);  
if (selectedDateTime > currentDate) {  
    alert("Please select a date in the past.");  
    return false;  
}
```

```
// Phone number validation
```

```
var phoneNumber = document.forms["myForm"]["phone"].value;  
var phoneNumberRegex = /^\d{10}$/;  
if (!phoneNumber.match(phoneNumberRegex)) {  
    alert("Please enter a 10-digit phone number.");  
    return false;  
}
```

```
// Email validation
```

```
var email = document.forms["myForm"]["email"].value;  
var emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;  
if (!email.match(emailRegex)) {  
    alert("Please enter a valid email address.");  
    return false;  
}
```

```
// Password confirmation
```

```
var confirmPassword = document.forms["myForm"]["confirmPassword"].value;  
if (password !== confirmPassword) {  
    alert("Password confirmation does not match.");  
    return false;  
}
```

```
// If all validations pass, the form is valid
```

```
        return true;
    }
</script>
</head>
<body>
    <form name="myForm" onsubmit="return validateForm()">
        <label for="username">Username:</label>
        <input type="text" id="username" name="username" required><br>

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

        <label for="country">Country:</label>
        <select id="country" name="country" required>
            <option value="Country1">Country1</option>
            <option value="Country2">Country2</option>
            <option value="Country3">Country3</option>
            <option value="Country4">Country4</option>
        </select><br>

        <label for="date">Date:</label>
        <input type="date" id="date" name="date" required><br>

        <label for="phone">Phone Number:</label>
        <input type="text" id="phone" name="phone" required><br>

        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required><br>

        <label for="confirmPassword">Confirm Password:</label>
        <input type="password" id="confirmPassword" name="confirmPassword"
required><br>

        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

Onmouseout and onmouseover

```
<!DOCTYPE html>
<html>
<body>
<h1>HTML DOM Events</h1>
<h2>The onmouseover Event</h2>
```

```

```

<p>The function bigImg() is triggered when the user moves the mouse pointer over the image.</p>

<p>The function normalImg() is triggered when the mouse pointer is moved out of the image.</p>

```
<script>
function bigImg(x) {
  x.style.height = "64px";
  x.style.width = "64px";
}
```

```
function normalImg(x) {
  x.style.height = "32px";
  x.style.width = "32px";
}
```

```
</script>
```

```
</body>
```

```
</html>
```