KONGU ENGINEERING COLLEGE, PERUNDURAI - 638 060
CONTINUOUS ASSESSMENT TEST – II
(Regulations 2020)
Answer Key

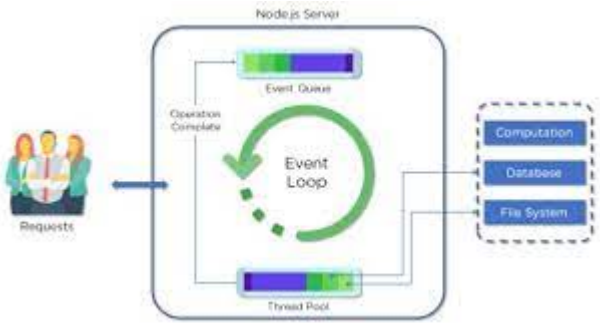
Month and Year :April 2023	Roll Number :
Programme : B.Tech. Branch : IT Semester : IV	Date : Time : 09.15 am to 10.45 am Duration : 1 ½ Hours Max. Marks : 50
Course Code : 20ITT44 Course Name : WEB TECHNOLOGY	

PART - A (10 × 2 = 20 Marks)			
ANSWER ALL THE QUESTIONS			

1.	Give a javascript snippet to display array values of single throw of a dice randomly.
	const diceValues=[1,2,,3,4,5,6];
	function getRandomElement(array)
	{
	Return array[Math.floor(Math.random() * array.length)];
	}
	const randomDiceValue = getRandomElement(diceValues);
	console.log(rondomDiceValue);
		CO2	K3

2.	Check whether the given roll number is valid(ex.21ITR001) or invalid using regular expression in javascript.
	
	if(roolno == "") {
	printError("rollnoErr", "Please enter your roll nuber");
	} else {
	var regex=^(21ITR/d/d/d)+$;
	
	if(regex.**test**(rollno) === false) {
	printError("rollnoErr", "Please enter a valid roll number");
	} else {
	printError("rollnoErr", "");
	rollnoErr = false;
	}
	}
		CO2	K3

3.	Mention the use of DOM and list out the property and methods of DOM collections. All html element elements will be considered as object.that allows programs and scripts to dynamically access and update the content, structure, and style of a document. Property and methods of DOM collections: length, item() and namedItem()	CO2	K1
4.	List out the different types of key and mouse events used in Javascript. **Mouse Events:** onmousedown, onmouseup, onmousemove,onmouseover,onmouseout **Key Events:** onkeydown, onkeypress, onkeyup	CO2	K1
5.	Denote the features and applications of NodeJs. **Features:** ● Node.js is an open source Run-time server environment ● Node.js is free ● Node.js allows you to run JavaScript on the server.	CO3	K1

	• Server-Side JavaScript Programming • Adv: Node.js - we can create our own server and port **Applications:** • Ebay • PayPal • Uber • Ebay		
6.	Draw the architecture and tell about the NodeJs process model.	CO3	K1
	 Node.js runs in a single process and the application code runs in a single thread and thereby needs less resources than other platforms. • Synchronous tasks happen **in order** — you must finish task one before moving on to the next. • Asynchronous tasks can be **executed in any order**, or even simultaneously.		
7.	Is end() and end in NodeJS same or different. If different, justify your answer. Both end() and end in NodeJS are different **end**: Event get emits when there is no available data to read **end()**: Method sends both the content of the response to the client and acknowledgement/signals to the server that the response (header and content) has been sent completely	CO3	K2

8.	How is code resuability enhanced in NodeJS.Give an example for displaying current date and time for the same.
	
	exports.getCurrentDateTime = function()
	{
	constcurrentDate = new Date() {
	constdateTimeString = currentDate.toLocalString();
	return dateTimeString;
	};
	
	const{ getCurrentDateTime}=require(‘./datetime’);
	constcurrentDateTime=getCurrentDateTime();
	console.log(currentDateTime);
		CO3	K3

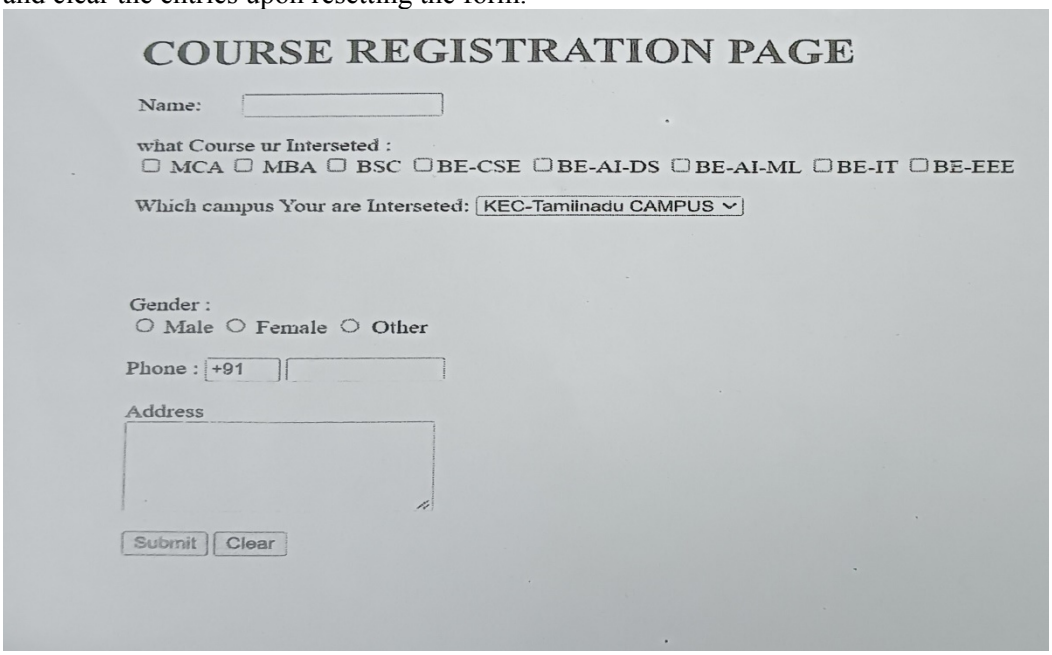
9.	Differentiate between SQL and NoSQL databases. SQL: • Relational Database Management System (RDBMS) • These databases have fixed or static or predefined schema • Follows ACID property • Eg:SQL NoSQL: • Non-relational or distributed database system • They have dynamic schema • Follows CAP(consistency, availability, partition tolerance) • Eg:MongoDb	CO3	K 2

10.	Specify the advantages and disadvantages of MongoDB. **ADVANTAGES:** • High scalability • High availability **DISADVANTAGES:** • No standardization rules • Limited query capabilities • Doesn't work as well with relational data • The learning curve is stiff for new developers • Open source options so not so popular for enterprises.		CO3	K 1

Part – B (3 × 10 = 30 Marks)

ANSWER ANY THREE QUESTIONS

11.		Design a course registration form as given below validating any four fields upon submission and clear the entries upon resetting the form.	(10)	CO2	K 3




```
<!DOCTYPE html>
<head>
<meta charset="UTF-8">
<title>REGISTRATION FORM</title>
<style>
   b {
         display:inline-block;
          width: 200px;
   }
   label {
   display:inline-block;
     width: 200px;
   }
   body {
  background-image: url("https://marketplace.canva.com/EAD2962NKnQ/2/0/1600w/canva-rainbow-gradient-pink-and-purple-zoom-virtual-background-_Tcjok-d9b4.jpg");
  background-repeat: no-repeat;
  background-size: 100%;
}
  </style>
<script>
```

```
function printError(elemId, hintMsg) {
   document.getElementById(elemId).innerHTML=hintMsg;
}
function validateForm() {
   var name = document.contactForm.name.value;
   var course = document.contactForm.email.value;
   var campus = document.contactForm.mobile.value;
   var gender = document.contactForm.gender.value;
   var city = document.contactForm.city.value;

   if(name == "") {
      printError("nameErr", "Please enter your name");
   } else {
      var regex = /^[a-zA-Z\s]+$/;
      if(regex.test(name) === false) {
         printError("nameErr", "Please enter a valid name");
      } else {
         printError("nameErr", "");
         nameErr = false;
      }
   }
   if(mobile == "") {
      printError("mobileErr", "Please enter your mobile number");
   } else {
      var regex = /^[1-9]\d{9}$/;
      if(regex.test(mobile) === false) {
         printError("mobileErr", "Please enter a valid 10 digit mobile number");
      } else{
         printError("mobileErr", "");
         mobileErr = false;
      }
   }
   if(gender == "") {
      printError("genderErr", "Please select your gender");
   } else {
      printError("genderErr", "");
      genderErr = false;
   }
if(city == "") {
      printError("cityErr", "Please enter your mobile number");
   } else {
      var regex = if(mobile == "") {
      printError("mobileErr", "Please enter your mobile number");
   } else {
      var regex = /^[1-9]\d{9}$/;
      if(regex.test(city) === false) {
         printError("cityErr", "Please enter a valid 10 digit mobile number");
      } else{
         printError("cityErr", "");
         cityErr = false;
      }
   }

      if(regex.test(mobile) === false) {
         printError("mobileErr", "Please enter a valid 10 digit mobile number");
      } else{
         printError("mobileErr", "");
         mobileErr = false;
      }
   }
```

		}; </script> </head> <body> <form *name*="contactForm" *onsubmit*="return **validateForm**()" *method*="post"> <Html> <body> Registration Page <form> <label>Firstname</label> <input type="text" name="firstname" size="15"/> <label> Course : </label> <select> <select value="Course">Course</option> <select value="MCA">MCA</option> <select value="MBA">MBA</option> <select value="Bsc">BSC</option> <select value="BE-CSE">BE-CSE</option> <select value="BE-AIDS">BE-AIDS</option> <select value="BE-AIML">BE-AIML</option> < select value="BE-AIML">BE-IT</option> < select value="BE-AIML">BE-EEE</option> </select> <label> Campus: </label> <select> <select value="KEC-Tamilnadu CAMPUS"> KEC-Tamilnadu CAMPUS</option> <select value="kec">kec</option> <select value="CIT">CIT</option> </label> Gender : </label> <input type="radio" name="male"/> Male <input type="radio" name="female"/> Female <input type="radio" name="other"/> Other <label> Phone : </label> <input type="text" name="country code" value="+91" size="2"/> <input type="text" name="phone" size="10"/> Address <textarea cols="80" rows="5" value="address"> </textarea> Submit Clear </form> </body> </html>			

12.		Write the javascript program to implement simple BMI calculator for healthcare assistant app using DOM handling functionalities. <!DOCTYPE html> <html>	(10)	CO2	K3

```
<head>
   <title>BMI Calculator</title>
</head>

<body>
   <h1>BMI Calculator</h1>
   <label for="weight">Weight (kg):</label>
   <input type="number" id="weight">
   <br>
   <label for="height">Height (cm):</label>
   <input type="number" id="height">
   <br>
   <button onclick="calculateBMI()">Calculate BMI</button>
   <br>
   <p id="result"></p>

   <script>
      function calculateBMI() {
         // Get weight and height input values
         var weight = document.getElementById("weight").value;
         var height = document.getElementById("height").value;

         // Check if weight and height are valid numbers
         if (isNaN(weight) || isNaN(height)) {
            document.getElementById("result").innerText = "Please enter valid weight and
height.";
            return;
         }

         // Calculate BMI
         var heightMeter = height / 100; // Convert height to meters
         var bmi = weight / (heightMeter * heightMeter); // BMI formula

         // Display the result
         document.getElementById("result").innerText = "BMI: " + bmi.toFixed(2);
      }
   </script>
</body>

</html>
```

13.	a)	Write the NodeJS program to create a server to handle the request from browser and provide the response based on the url properties and buffer contents. **GET METHOD:** <html> <head> <title></title> </head> <body> <form action=http://localhost:8000/login method=”GET”> <input type=”text” name=”username” value””/> <input type=”text” name=”email” value””/> <input type=”submit” name=”login” value=”Login”/> </form> </body> </html> http=require(‘http’);	(6)	CO3	K2

```
url=require('url');
querystring=require('querystring');
function onRequest(request,response)
{
  var path=url.parse(request.url).pathname;
console.log('Request for '+patg+' received. ');
  var query=url.parse(request.url).query;
  console.log(query);
var name=querystring.parse(query)["username"];
var email=querystring.parse(query)["email"];
response.write('Hello '+name+' , your email id '+email+'
response.end();
}
http.createServer(onrequest).listen(8000);
console.log('Server has Started..');
```

POST METHOD:

```
<html>
<head>
<title></title>
</head>
<body>
<form action=http://localhost:8000/login method="post">
<input type="text" name="username" value""/><br/>
<input type="text" name="email" value""/><br/>
<input type="submit" name="login" value="Login"/>
</form>
</body>
</html>
var http=require('http');
var querystring=require('querystring');
var qs,name,email;
http.createServer(function(req,res)
  {
     var data1= ' ';
req.on('data', function(chunk)
      {
          Console.log(chunk);
           Data1+=chunk;
console.log("Data in string format: "+data1);
      });
req.on('end', function()
       {
qs=querystring.parse(data1);
           console.log(qs);
           name=qs['username'];
           email=qs['email'];
res.write("Hello "+name+", your email id "+email+" has been
res.end();
       });
}).listen(7777);
console.log("Server starts");
```

b) Outline the different in-built and custom event handling performed in Node-JS. (4)

Custom events:

- events: Node js built in module that is used to create, fire and listen for own events
- To include the built-in Events module use the require() method.
- EventEmitter: Event handler can be assigned to events using event emitter object.
- emit(): Used to fire/invoke an event
- on(): Used to create event with 2 args - Event name & Function to be

		executed on event **In-built events:** ○ data ○ end ○ error ○ finish			

14. i.	Write mongodb queries to perform the following operations: Insert the following information about the product like product id,productName,salestype[retail,wholesale] and price. db.products.insertOne({ productid: productName: “Example Product”, salestype: “retail”, price: 9.99 });	(10)	CO3	K3
ii.	Sort the product information based on the sale type and display the details. db.products.find().sort({ salestype: 1 });			
iii.	Update the product price based on product id db.products.updateOne({ productsId: 1], { $set: { price: 19.99 } })			
iv.	Display all the product details db.products.find();			
v.	Delete product details based on product id db.products.deleteOne({ productid: 1 });			

Bloom’s Taxonomy Level	Remembering (K1)	Understanding (K2)	Applying (K3)	Analysing (K4)	Evaluating (K5)	Creating (K6)
Percentage	16.67	23 .33	60	--	--	--