Web Qbank

RAMAIAH Institute of Technology

Department of Computer Science & Engineering

QUESTION BANK FOR VI SEMESTER (Term: Mar-Jul 2022)

Web Technologies Laboratory (CSL68)

I.A. Marks : 50 **Exam Hours: 03**
Credits : 0:0:1 **Exam Marks: 50**

1.	(a) Write a function translate() that will translate a text i.e, double every consonant and place an occurrence of "o" in between. For example, translate("this is fun") should return the string "tothohisos isos fofunon". (b) Using Node.js Express and Mongo, implement a program to accept USN, Name, Subject_code, CIE marks and store the information in a database and display students whose CIE<20
2.	(a) Write a java script program to convert month number to month name using closures. ● If the user enters a number less than 1 or greater than 12 or a non-number, have the function write "Bad Number" in the monthName field. ● If the user enters a decimal between 1 and 12 (inclusive), strip the decimal portion of the number. (b) Write a node.js Express and Mongo program to accept Student_name,USN,semester,exam_fee from web page and delete all the students who have not paid exam fees.
3.	(a) Write a javascript to implement a Calculator using prototype, which has add, subtract and getAnswer functions which supports chaining, that means we should be able to do new Calculator(2).add(2).add(2).subtract(3).getAnswer() to get 3 as the answer. (b) Write a node.js Express and Mongo program to create a 'HR' database with the collection 'employees' having the fields like emp_name,email,phone, hire_date, job_title,salary. Accept these fields information from a web page and store it in the database and display all the employee details whose salary>50000.
4.	(a) Write an REACT program to print Name, Address and Company of an Employee. When you Click on the CHANGE button, the name and address should be changed. (b) Write a Node.js program using Express framework and create an on-line training site with three pages of content: Home, Registration, Announcements & Contact. Use routing to swap between them..
5.	(a) Write a java script function named **pluralize** that: ● takes 2 arguments, a noun and a number. ● returns the number and pluralized form, like "5 cats" or "1 dog". ● Make it handle a few collective nouns like "sheep" and "geese". (b) Write a Node.js Express and Mongo program to accept 'Student' information viz. Name, USN, Dept, Grade from a web page and store the information in a database and update Student grade with the name specified by the user and display the results.
	(a) Write an REACT program which accepts the Name from the form. As you type, it updates the Name in the page with an h1 tag. (b) Write a Node.js program using Express framework to display different branch information offered in an Engineering College with different background color and fonts (Note: Use Routing, Min: 3 branches)

6.	(a) Design a Student Form using HTML5 which has following fields a) Name : Required must be characters b) Email : Validation placeholder: please enter valid email address c) Phone : accept numbers in the following format (080-555-5555) d) Semester : For the range 1 to 8 e) Branch :Data list f) Website :Required pattern of the form-http:// (b) Create an Exam Management system using the MERN Stack framework for creating student databases and displaying students who have secure 'S' grades. (Use Appropriate fields)
7.	(a) Write an npm script having a function vowelCount() that takes a string as input and counts the number of occurrences of each vowel in the string. (Hint: run the program through **npm start**) For. Eg. Input : vowelCount('Le Tour de France') Output: a, e, i, o, and u appear, respectively, 1, 3, 0, 1, 1 times (b) Using node.js Express and Mongo to implement a 'FinalYears' database which accepts 'USN','Name' and 'Company_name' (by campus selection) as fields and s ubtore it in a database. Display the list of students who are selected for 'Infosys'
8.	(a) Write a Node.js program using Express framework and create an on-line training site with three pages of content: Home, Registration, Announcements & Contact. Use routing to swap between them. (b) Write a Node.js Express and Mongo program to accept USN, Name, Branch, Semester, from the web page and display all the students who belong to 6th Semester and CSE branch.
9.	(a) Write a node.js Express program to create a custom middleware functions for i. Logger ii. No. of time the visitor visited the website (b) Write a Node.js Express and Mongo program to accept the fields 'ID', 'Title', 'Name', and 'branch' of a faculty and store it in the database. Display all the faculty who belong to the "CSE" branch and Title is "PROFESSOR".
10.	(a) Create a web page with the following characteristics using BOX Model ● h1's have 1px red solid borders, background color #D18C1D, and 10px of space between the content and the border (padding) ● List items have 15px extra space around them (margin) and background color #C0A9DB ● Paragraphs are contained in 600px by 400px boxes with 2px black dotted borders and background color #D1D631 (b) Create an Attendance Management system using Nodejs Express and Mongo for creating student database and display student's whose attendance is below 75% (Use Appropriate fields)
11.	(a) Write a java script program to implement Stack and Queue using **modules** (b) Create an Exam Management system using Node Js Express and Mongo for creating a student database and display Not Eligible Student List based on the Marks<20. (Use Appropriate fields)

Marks Distribution

Conduction and Result	Write Up	Execution	Viva	Change of Program	Total
Part – a	8	15 Marks	7 Marks	-10 Marks	50 Marks
Part – b		20 Marks			

All programs Here https://github.com/anishgowda21/6th-sem-labs/blob/main/weblab/ExamProblems

1 a.

```
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
    <title>Document</title>
  </head>
  <body>
  Enter text here:<input type="text" required id="inText"/><br/>
    <button onclick="sbtBtn()">Translate</button><br />
    <div id="result"></div>

    <script>
      function sbtBtn() {
        var text = document.getElementById("inText").value;
        console.log(text);
        var vowels = ["a", "e", "i", "o", "u"];
        var result = "";
        for (let i = 0; i < text.length; i++) {
          element = text[i];
          if (element === " ") {
            result += " ";
          } else if (vowels.includes(element)) {
            result += element;
          } else {
            result += element + "o" + element;
          }
        }
        document.getElementById("result").innerHTML = result;
      }
    </script>
  </body>
</html>
```

1b.

```
var express = require('express');
var MongoClient = require('mongodb').MongoClient;
var app = express();

app.get('/', function (req, res) {
  res.sendFile(__dirname + '/1b.html')
})

app.get('/data', function (req, res) {
  var usn = req.query.usn
  var name = req.query.name
  var subcode = req.query.subcode
  var cie = parseInt(req.query.marks)
  var obj = { "usn": usn, "name": name, "subcode": subcode, "cie":
cie };
  MongoClient.connect('mongodb://127.0.0.1:27017/nodedb', function
(err, db) {
    if (!err) {
      db.collection('student').insertOne(obj, function (err, db) {
        if (!err) {
          console.log("Successful document insertion")
          res.send("<a href='/'>Insert More data</a><br><br><a
href='/show'>Show Students with CIE less than 20</a>")
        }
        else {
          console.log("Unsuccessful!")
          db.close()
        }
      })
    }
    else {
      console.log("Couldn't connect to db")
      db.close()
    }
  })
})

app.get('/show', function (req, res) {
  MongoClient.connect('mongodb://127.0.0.1:27017/nodedb', function
(err, db) {
    if (!err) {
      console.log("Mongo successfully connected in show")
      var disp = db.collection('student').find({ "cie": { $lt: 20
} })
      res.write("<h1>Students below 20 in CIE</h1>")
```

```
      disp.each(function (err, item) {
        if (item != null) {

        }
      })
    }
  })
})
app.listen(5000)
```

HTML File:

```
<!doctype html>
<html>
<head>
      <title>Form</title>
</head>
<body>
      <form method="GET" action="data">
            USN:<input type="text" name="usn" id="usn"><br><br>
            Name:<input type="text" name="name" id="name"><br><br>
            Subject Code:<input type="text" name="subcode"
id="subcode"><br><br>
            CIE Marks:<input type="text" name="marks"
id="marks"><br><br>
            <input type="submit">
      </form>
</body>
</html>
```

2a.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Closures</title>
</head>
<body>
    <script>
        function call(){
            var num = parseFloat(prompt("Enter month num:",0));
            if(num>=1 && num<=12){
```

```
                num = Math.trunc(num);
                var res = (function(){
                    var  months = ["January", "February", "March",
"April", "May", "June", "July", "August", "September", "October",
"November", "December"];
                    return months[num-1];
                })();
            }else{
                return "Bad Number";
            }
            return res;
        }
        var result = call();
        document.write("<h1>",result,"</h1>");
    </script>
</body>
</html>
```

2b.

Index.html

<html>

<body>

<form action="process_get" method="GET">

Name: <input type="text" name="name">

USN: <input type="text" name="usn">

Semester: <input type="text" name="sem">

Exam Fees: <input type="text" name="examfee">

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

</form>

</body>

</html>

Index.js

```
var express = require('express');
var app = express();
var MongoClient = require('mongodb').MongoClient;
app.get('/',function(req,res){
res.sendFile(__dirname+"/"+"index.html");
})
app.get('/process_get',function(req,res){
response = {
usn : req.query.usn,
name : req.query.name,
sem : req.query.sem,
examFee : req.query.examfee
};
console.log(response);
MongoClient.connect('mongodb://127.0.0.1:27017',function(err,client){
if(err) throw err;
var db = client.db('mydb');
var collection = db.collection('student2b');
 collection.insert(response);
await collection.deleteMany({"examFee" : "No"});
collection.find({}).toArray(function(err,results){
//console.dir(results);
```

```
if(err) throw err
res.render('disp.ejs',{students:results})
client.close();
});
});
});
var server = app.listen(5000,function(){
var host = server.address().address;
var port = server.address().port;
console.log("Example app listening at http://%s:%s",host,port);
});
```

Disp.ejs

```
Student Information
<% for(var i=0;i<students.length;i++){%>
<p><%= "USN: "+students[i].usn %></p>
<p><%= "Name: "+students[i].name %></p>
<p><%= "Semester: "+students[i].sem %></p>
<p><%= "Exam Fees: "+students[i].examFee %></p>

<% } %>
```

3a.

```
<!doctype html>
<html>
<head>
      <script type="text/javascript">
            function calculator(num)
            {
                  this.num=num
            }
            calculator.prototype.Add=function(n)
            {
                  this.num=this.num+n
                  return this
            }
            calculator.prototype.subtract=function(n)
            {
                  this.num=this.num-n
                  return this
            }
            calculator.prototype.getAnswer=function()
            {
                  return this.num
            }

            var ans = new
calculator(2).Add(2).Add(2).subtract(4).getAnswer()
            console.log(ans)
      </script>
</head>
</html>
```

5a.

```
function pluralize(noun, number) {
    if (number != 1 && noun != 'sheep' && noun != 'geese') {
        return number + ' ' + noun + 's';
    } else {
        return number + ' ' + noun;
    }
}
console.log('I have ' + pluralize('cat', 0));
console.log('I have ' + pluralize('cat', 1));
console.log('I have ' + pluralize('cat', 2));
```

5b.

```
var express=require('express')
var app=express()
var MongoClient=require('mongodb').MongoClient
```

```
app.get('/',function(req,res) {
      res.sendFile(__dirname+'/5b.html')
})

app.get('/data',function(req,res) {
      var obj={"name":req.query.name,"usn":req.query.usn}

MongoClient.connect("mongodb://127.0.0.1:27017/nodedb",function(er
r,db) {
            if(!err) {
                  console.log('Connected to db')
                  db.collection('b5').insertOne(obj,function(err,db)
{
                        if(!err) {
                              res.end("<p>Successful
insertion</p><br><a href='/'>Insert</a><br><a
href='/update'>Update Data</a>")
                        }
                  })
            }
      })
})

app.get('/update',function(req,res) {
      res.sendFile(__dirname+'/5b_update.html')
})

app.get('/result
s',function(req,res) {
      MongoClient.connect("mo
ngodb://127.0.0.1:27017/nodedb",function(err,db) {
            if(!err) {

db.collection('b5').updateOne({"name":req.query.name},{$set:{"usn"
:req.query.usn}},function(err,db) {
                        if(!err) {
                              console.log('Successful Update')
                        }
                  })
                  var cur=db.collection('b5').find()
                  res.write("<h1>Display DB</h1>")
                  cur.each(function(err,item) {
                        if(item!=null) {
                              res.write(item.name)
                              res.write("<br>")
                              res.write(item.usn)
                              res.write("<br><br>")
                        }
```

```
                })
            }
      })
})

app.listen(5000)
```

5b.html:

```
<!doctype html>
<html>
<head>
      <title>Form</title>
</head>
<body>
      <form method="GET" action="data">
            USN:<input type="text" name="usn" id="usn"><br><br>
            Name:<input type="text" name="name" id="name"><br><br>
            Subject Code:<input type="text" name="subcode"
id="subcode"><br><br>
            CIE Marks:<input type="text" name="marks"
id="marks"><br><br>
            <input type="submit">
      </form>
</body>
</html>
```

5b_update.html:

```
<html>
<body>
      <form method="GET" action="results">
            Enter Name to change USN: <input type="text" name="name"
id="name"><br>
            New USN: <input type="text" name="usn" id="usn"><br>
            <input type="submit">
      </form>
</body>
</html>
```

6b.

```
var express = require('express');
var app = express();

app.get('/', function(req, res) {
 res.send('Welcome to the Engineering College!');
});

app.get('/branches', function(req, res) {
 res.send('The different branches offered in our college
```

```
are:<br>1.CivilEngineering<br>2.Mechanical
Engineering<br>3.Electrical and Electronics
Engineering<br>4.Computer Science');
});

app.get('/branches/civil', function(req, res) {
res.send('<p
style="background-color:lightblue;font-family:verdana;">Civil
Engineering </p>');
});

app.get('/branches/mechanical', function(req, res) {
  res.send('<p style="background-color:lightgreen;">Mechanical
engineering.</p>');
});

app.get('/branches/electrical', function(req, res) {
  res.send('<p style="background-color:red;">Electrical
engineering.</p>');
});
app.get('/branches/computers',function(req,res){
 res.send('<p style="background-color:black;">Computer
science.</p>');
});
var server = app.listen(3000, function() {
  console.log("Example app listening at ", port)
});
```

7a.

```
<!doctype html>
<html>
<head>
     <h3>STUDENT INFORMATION FORM</h3>
</head>
<body>
     <form name="data">
          Name : <input type="text" pattern="[a-zA-Z]+"><br>
          Email : <input type="email"
placeholder="abc@xyz.com"><br>
          Phone : <input type="text"
pattern="[0-9]{3}-[0-9]{3}-[0-9]{4}"><br>
          Semester : <input type="number" min=1 max=8><br>
          Branch : <input list="dl">
          <datalist id="dl">
               <option value="CSE"/>
               <option value="EC"/>
               <option value="ISE"/>
          </datalist><br>
```

```
          Website : <input type="url"><br>
          <input type="submit">
     </form>
</body>
</html>
```

7b.

8a.

```
function vowelCount(str) {
    let vowels = ['a', 'e', 'i', 'o', 'u'];
    let count = {};
    for (let i = 0; i < str.length; i++) {
        if (vowels.includes(str[i])) {
            if (count[str[i]]) {
                count[str[i]]++;
            } else {
                count[str[i]] = 1;
            }
        }
    }
    return count;
}

let text = 'Le Tour de France';
let count = vowelCount(text);

console.log("Vowel Frequency in '" + text + "' is: " +
JSON.stringify(count));
```

8b.

```
var express=require('express')
var app=express()
var MongoClient=require('mongodb').MongoClient

MongoClient.connect("mongodb://127.0.0.1:27017/nodedb",function(er
r,db) {
     if(!err) {
          console.log("Connected to DB")
          app.get('/',function(req,res) {
               res.sendFile(__dirname+'/8b.html')
          })

          app.get('/data',function(req,res) {
               var
obj={"usn":req.query.usn,"name":req.query.name,"company":req.query
.comp}
```

```
db.collection('student_8b').insertOne(obj,function(err,db) {
                    if(!err) {
                    console.log("Document successfully Inserted")
                    }
                })
                res.end("<p>Document Successfully
Inserted</p><br><a href='/'>Insert</a><br><a
href='display'>Display</a>")
          })

          app.get('/display',function(req,res) {
                var
cur=db.collection('student_8b').find({"company":"infosys"})

                res.write('<h1>Infosys selected students</h1>')
                cur.each(function(err,item) {
                    if(item!=null) {
                        res.write("Name : "+item.name+"<br>")
                        res.write("USN : "+item.usn+"<br>")
                        res.write("Company :
"+item.company+"<br><br>")
                    }
                })
          })
          app.listen(5000)
     }
})
```

8b.html:

```
<!doctype html>
<html>
<head>
     <h3>FINAL YEAR INFORMATION</h3>
</head>
<body>
     <form method="GET" action="data">
          USN: <input type="text" name="usn" id="usn"><br>
          Name: <input type="text" name="name" id="name"><br>
          Company Name: <input type="text" name="comp"
id="comp"><br>
          <input type="submit"><br>
     </form>
     <a href='display'>Display</a>
</body>
</html>
```

9b.

```
var express=require('express')
var app=express()
var MongoClient=require('mongodb').MongoClient

MongoClient.connect('mongodb://127.0.0.1:27017/nodedb',function(er
r,db) {
     if(!err) {
          console.log('Connected to DB')
          app.get('/',function(req,res) {
               res.sendFile(__dirname+'/9b.html')
          })
          app.listen(5000)
          app.get('/data',function(req,res) {
               var
obj={"username":req.query.uname,"branch":req.query.branch,"sem":re
q.query.sem}

db.collection('student_9b').insertOne(obj,function(err,db) {
                    if(!err) {
                         console.log('Document Inserted')
                         res.end("<p>Document Successfully
Inserted</p><br><a href='/'>Insert</a><br><a
href='display'>Display</a>")
                    }
               })
          })

          app.get('/display',function(req,res) {
               console.log('Display')
               var
cur=db.collection('student_9b').find({$and:[{"branch":"CSE"},{"sem
":"6"}]})
               res.write('<h1>6th Sem CSE students</h1>')
               cur.each(function(err,item) {
                    if(item!=null) {

res.write("Username"+item.username+"<br>")
                         res.write("Branch :
"+item.branch+"<br>")
                         res.write("Sem : "+item.sem+"<br><br>")
                    }
               })
          })
     }
})
```

9b.html:

```
<!doctype html>
```

```
<html>
<head>
	<h3>STUDENT INFORMATION</h3>
</head>
<body>
	<form method="GET" action="data">
		Username: <input type="text" name="uname"
id="uname"><br>
		Branch: <input type="text" name="branch"
id="branch"><br>
		Semester: <input type="text" name="sem" id="sem"><br>
		<input type="submit"><br>
	</form>
	<a href='display'>Display</a>
</body>
</html>
```

10a.

```
	var express=require('express')
	var app=express()

	var count=0;
	function logger(req,res,next) {
		console.log("Logged in")
		count++
		next()
	}

	app.use(logger)

	var visit=function(req,res,next) {
		res.visit=count
		console.log("visited : "+count)
		next()
	}

	app.use(visit)

	app.get('/',function(req,res) {
		res.send("<h3>Visited : "+res.visit+"</h3>")
	})

	app.listen(5000)
```

11a.

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

```
<head>
	<style>
h1{background-color:#D18C1D;border-width:1px;border-style:solid;border-color:red;padding:10px;}
	li{margin:15px;background-color:#C0A9DB;}
p{height:400px;width:600px;border-style:dotted;border-width:2px;background-color:#D1D631;}
	</style>
</head>
<body>
	<h1>TITLE</h1>
	<ul>
		<li>list element 1</li>
		<li>list element 2</li>
	</ul>
	<p>This is a paragraph</p>
</body>
</html>
```

12a.

```
let stack = require("./stack");
let queue = require("./queue");

let Stack = new stack();
let Queue = new queue();

console.log("Stack");
Stack.push(1);
Stack.push(2);
console.log("Current Stack");
Stack.print();
console.log("Top");
console.log(Stack.return_top());
console.log("Pop");
console.log(Stack.pop());
Stack.clear();
console.log("Clear");
Stack.print();

console.log("Queue");
Queue.enqueue(1);
Queue.enqueue(2);
Queue.enqueue(3);
console.log("Current Queue");
Queue.print();
console.log("Dequeue");
```

```
console.log(Queue.dequeue());
console.log("Clear");
Queue.clear();
Queue.print();
```

stack.js:

```
function Stack() {
  this.stack = [];
}

Stack.prototype.push = function(element) {
  this.stack.push(element);
};

Stack.prototype.pop = function() {
  let element = this.stack.pop();
  return element;
};

Stack.prototype.print = function() {
  if (!this.stack.length) {
    console.log("Empty");
    return;
  }
  let s = [].concat(this.stack);
  s.reverse();
  s.forEach(function println(item) {
    console.log(item);
  });
};

Stack.prototype.return_top = function() {
  return this.stack[this.stack.length - 1];
};

Stack.prototype.clear = function() {
  this.stack = [];
};

module.exports = Stack;
```

queue.js:

```
function Queue() {
  this.queue = [];
}

Queue.prototype.enqueue = function(element) {
```

```
  this.queue.push(element);
};

Queue.prototype.dequeue = function() {
  let element = this.queue.shift();
  return element;
};

Queue.prototype.clear = function() {
  this.queue = [];
};

Queue.prototype.print = function() {
  if (!this.queue.length) {
    console.log("Queue is Empty");
    return;
  }
  let result = "";
  this.queue.forEach(function println(item) {
    result += item + " ";
  });
  console.log(result);
};

module.exports = Queue;
```