# EXPRESS

1. **cookies.js**

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

app.get('/', function(req, res){
   res.cookie('name', 'vidyavardhini').send('cookie set'); //Sets name =
express
});

app.listen(3000);
```

2. **express1.js**

```
const express=require('express')
const app=express()
const port=3000

app.get('/',(req,res)=>{
res.send('hello world')
})


app.post('/about',(req,res)=>{
res.send('This is VCET')
})

app.listen(port,()=>{
console.log('app listening on port 3000')
})
```

3. **expressrouter.js**

```
var express = require('express');
var app=express()
var router = express.Router();
const port = 3000

// Home page route.
router.post('/', function (req, res) {
  res.get('Wiki home page');
```

```
})

// About page route.
router.get('/about', function (req, res) {
  res.send('About this wiki');
})

app.listen(port, () => {
  console.log(`Example app listening at http://localhost:${port}`)
})


module.exports = router;
```

### 4. router1.js

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

app.get('/hello', function(req, res){
   res.send("Hello World!");
});

app.get('/hello1', function(req, res){
   res.send("You just called the post method at '/hello'!\n");
});

app.get('/hello2', function(req, res){
   res.send("You just called the put method at '/hello'!\n");
});

app.listen(3000);
```

### 5. session.js

```
var express = require('express');
var cookieParser = require('cookie-parser');
var session = require('express-session');

var app = express();

app.use(cookieParser());
app.use(session({secret: "Shh, its a secret!",

}));

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

```
   if(req.session.page_views){
      req.session.page_views++;
      res.send("You visited this page " + req.session.page_views );
   } else {
      req.session.page_views = 1;
      res.send("Welcome to this page for the first time!");
   }
});


app.listen(3000);
```

### 6. simpleapp.js

```
const express = require('express')
const app = express()
const port = 3000


app.get('/', (req, res) => {
  res.send('Hello World!')
})


app.listen(port, () => {
  console.log(`Example app listening at http://localhost:${port}`)
})
```

# NODE JS

1. **app.js**

```
const http = require('http')
http.createServer(function(req,res){
res.write("Hello");

res.end()

}).listen(3000);
```

2. **block.js**

```
const fs = require('fs');
const data = fs.readFileSync('mongo.js'); // blocks here until file is read
console.log(data);
console.log("hello");
```

3. **blocking.js**

```
const fs = require('fs');
const data = fs.readFileSync('mongo.js'); // blocks here until file is read
console.log(data);
console.log("hello");
```

4. **c.js**

```
var http = require('http');
var options = {
   host: 'localhost',
   port: '8081',
};
var callback = function(response) {
   var body = '';
   response.on('data', function(data) {
      body += data;
   });

   response.on('end', function() {
```

```
      console.log(body);
   });
}
```

5. **client.js**

```
var net = require("net");
var client = new net.Socket();
var host="localhost";
var port=9000;
client.connect( port, host, () => {
console.log("client connected to ${host}:${port}");
client.write("Hello, I am ${client.address().address}");
});
client.on('data', (data) => {
    console.log(`Client received: ${data}`);
});
```

6. **client1.js**

```
var net = require("net");
var client = new net.Socket();
var host="localhost";
var port=9000;
client.connect( port, host, () => {
console.log("client connected to ${host}:${port}");
client.write("Hello, I am ${client.address().address}");
});
client.on('data', (data) => {
    console.log(`Client received: ${data}`);
});
```

7. **connection.js**

```
const sql=require('mssql');

const sqlConfig = {
  user: 'sa',
  password: 'chandan',
  database: 'vcet',
  port:55360,
  server: '127.0.0.1',
```

```
 pool: {
    max: 10,
    min: 0,
    idleTimeoutMillis: 30000
  },
  options: {
    encrypt: false, // for azure
    trustServerCertificate: false // change to true for local dev / self-
signed certs
  }
}



var a= async()=>{
try{
var q= await sql.connect(sqlConfig);
const result =await sql.query("select * from inft");
console.log(result.recordsets);
}catch(err)
{console.log(err);
}
}
a();
```

8. **cv.js**

```
console.log("Hello, World!");
var http = require("http");
http.createServer(function(request,response){
response.writeHead(200, {'Content-Type': 'text/plain'});
response.end('Hello World\n');
}).listen(8081);
```

9. **event.js**

```
var events = require('events');
var eventEmitter = new events.EventEmitter();
var connectHandler = function connected() {
   console.log('connection succesful.');
     // Fire the data_received event
   //eventEmitter.emit('data_received');
}
// Bind the connection event with the handler
eventEmitter.on('connect', connectHandler);
```

```
 // Bind the data_received event with the anonymous function
eventEmitter.on('data', function() {
   console.log('data received succesfully.');

});

eventEmitter.on('terminate', function() {
   console.log('program terminated');

});

// Fire the connection event
eventEmitter.emit('connect');
eventEmitter.emit('data');
eventEmitter.emit('terminate');

console.log("Program Ended.");
```

### 10.event1.js

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

var eventEmitter =new events.EventEmitter();
var connectHandler=function connect()
{
console.log('connection successful');
eventEmitter.emit('received');
}

var receiveHandler=function receive()
{
console.log('received successful');
}


eventEmitter.on('received',receiveHandler);


eventEmitter.on('connection',connectHandler);
eventEmitter.emit('connection');
```

### 11.eventloop.js

```
console.log("This is the first statement");
console.log("This is the fourth statement");
```

```
setTimeout(function(){
    console.log("This is the second statement");
}, 1000);

console.log("This is the third statement");
```

### 12. expresstry.js

```
const express = require('express')
var MongoClient = require('mongodb').MongoClient;
const app = express()
const port = 3050
var url = "mongodb://localhost:27017/";

app.get('/', (req, res) => {
  res.send('Hello World!')
})


app.get('/getdata', (req, res) => {
MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("vcet");
  var myobj = { name: "VCET" };
  dbo.collection("inft").find(myobj).toArray(function(err, res) {
    if (err) throw err;
    console.log(res);


  db.close();
  });
});
  res.send('1 record inserted')
})

app.get('/putdata', (req, res) => {
MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("vcet");
  var myobj = { name: "VCET", address: "KT marg Vasai" };
  dbo.collection("inft").insertOne(myobj, function(err, res) {
    if (err) throw err;
    console.log("1 document inserted");
    db.close();
  });
});
  res.send('1 record inserted')
```

```
})
```

```
app.listen(port, () => {
  console.log(`Example app listening at http://localhost:${port}`)
})
```

### 13.fetch.js

```
fetch('localhost:8080')
  .then(
    function(response) {
      if (response.status !== 200) {
        console.log('Looks like there was a problem. Status Code: ' +
          response.status);
        return;
      }

      // Examine the text in the response
      response.json().then(function(data) {
        console.log(data);
      });
    }
  )
  .catch(function(err) {
    console.log('Fetch Error :-S', err);
  });
```

### 14. fetch1.js

```
fetch('localhost:8080')
  .then(
    function(response) {
      if (response.status !== 200) {
        console.log('Looks like there was a problem. Status Code: ' +
          response.status);
        return;
      }
```

```
      // Examine the text in the response
      response.json().then(function(data) {
        console.log(data);
      });
    }
  )
  .catch(function(err) {
    console.log('Fetch Error :-S', err);
  });
```

### 15.file.js

```
var http = require('http');
var fs = require('fs');
http.createServer(function (req, res) {
  //Open a file on the server and return its content:

fs.appendFile(
'mongo.js',
'New access log entry',
error => {
if (error) {
console.error('error writing!', error)
}

fs.readFile('mongo.js', function(err, data) {
    res.writeHead(200, {'Content-Type': 'text/html'});
    res.write(data);
    return res.end();
  });

})

}).listen(8080);
```

### 16.file1.js

```
const fs=require('fs');
const data=fs.readFile('mongo.js',(err,data)=>{
if(err) throw err;
console.log(data);


});
```

```
console.log("hello");
```

### 17.fire.js

```
const fs = require('firebase-admin');

const serviceAccount = require('./google-services.json');

fs.initializeApp({
 credential: fs.credential.cert(serviceAccount)
});
```

### 18. fire1.js

```
var admin = require("firebase-admin");

var serviceAccount = require("./test-dad7d-firebase-adminsdk-sz83s-
bd946a5f85.json");

admin.initializeApp({
  credential: admin.credential.cert(serviceAccount)
});

const db = admin.firestore();
const usersDb = db.collection('chandan');

db.collection("chandan").get().then((querySnapshot) => {
    querySnapshot.forEach((doc) => {
        console.log(`${doc.id} => ${doc.data()}`);
    })});


db.collection("chandan").where("name", "==", "chandan")
    .get()
    .then(function(querySnapshot) {
        querySnapshot.forEach(function(doc) {
            // doc.data() is never undefined for query doc snapshots
            console.log(doc.id, " => ", doc.data());
        });
    })
    .catch(function(error) {
        console.log("Error getting documents: ", error);
    });
```

```
//console.log(liam);
/*  liam.set({
 first: 'Liam',
 last: 'Ragozzine',
 address: '133 5th St., San Francisco, CA',
 birthday: '05/13/1990',
 age: '30'
});
*/



//console.log(liam);
```

### 19.fire2.js

```
var admin = require("firebase-admin");

var serviceAccount = require("./test-dad7d-firebase-adminsdk-sz83s-
bd946a5f85.json");

admin.initializeApp({
  credential: admin.credential.cert(serviceAccount)
});

const db=admin.firestore();
const userdb =db.collection('chandan');
const record= userdb.doc('vcet1');

const d= db.collection("chandan").where("name","==","chandan")
.get()
.then(function(query){
query.forEach(function(doc)
{console.log(doc.id,":",doc.data())})})
.catch(function(error){
console.log("error");
});
```

## 20.https.js

```
var http = require('http');
var fs = require('fs');
var url = require('url');

http.createServer( function (request, response) {
   fs.readFile("index.html", function (err, data) {
      if (err) {
         console.log(err);

         response.writeHead(404, {'Content-Type': 'text/html'});
      } else {
         response.writeHead(200, {'Content-Type': 'text/html'});
         response.write(data.toString());
      }

      response.end();
   });
}).listen(8081);

console.log('Server running at http://127.0.0.1:8081/');
```

## 21.mongo.js

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("vcet");
  var myobj = { name: "VCET", address: "KT marg Vasai" };
  dbo.collection("inft").insertOne(myobj, function(err, res) {
    if (err) throw err;
    console.log("1 document inserted");
    db.close();
  });
});
```

## 22.mongo2.js

```
var mongodbclient=require('mongodb').MongoClient;
```

```
var url="mongodb://localhost:27017/";

mongodbclient.connect(url,function(err,db){

if(err) throw err;

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

var myobj={name:"chandan",address:"ratnagiri"};

dbo.collection("inft").findOne({},function(err,res){
if(err) throw err;
console.log(res);
db.close();
});

})
```

### 23.net1.js

```
var net=require ("net");
var server = net.createServer();
server.on('connection', handleConnection);
server.listen(9000, function() {
  console.log('server listening to %j', server.address());
});
function handleConnection(conn) {
  var remoteAddress = conn.remoteAddress + ':' + conn.remotePort;
  console.log('new client connection from %s', remoteAddress);
 conn.on('data', onConnData);
 function onConnData(d) {
    console.log('connection data from %s: %j', remoteAddress, d);
    conn.write(d);
  }
}
```

### 24.nonblock.js

```
const fs = require('fs');
fs.readFile('mongo12333333.js', (err, data) => {
  if (err) throw err;
  console.log(data);
});
console.log("hello"); // will run before console.log
```

### 25.nonblockwrite.js

```
const fs = require('fs');

const content = 'Some content!'

fs.writeFile('mongo.js', content, err => {
  if (err) {
    console.error(err)
    return
  }
  //file written successfully
})



fs.readFile('mongo.js', (err, data) => {
  if (err) throw err;
  console.log(data);
});
console.log("hello"); // will run before console.log
```

### 26.read1.js

```
var fs=require("fs");
var data='';
var counter=0;
var readStream = fs.createReadStream("E:\\books\\bigdata.pdf");
readStream.setEncoding('UTF-8');
readStream.on('data',function(chunk){
data +=chunk;
counter=counter+1;
console.log(counter);
});
readStream.on('end',function(){
//console.log(data);
console.log("ended")});
```

### 27. readstream.js

```
var fs = require("fs");
var data = '';

// Create a readable stream
var readerStream = fs.createReadStream('index.html');

// Set the encoding to be utf8.
readerStream.setEncoding('UTF8');

// Handle stream events --> data, end, and error
readerStream.on('data', function(chunk) {
   data += chunk;
});

readerStream.on('end',function() {
   console.log(data);
});

readerStream.on('error', function(err) {
   console.log(err.stack);
});

console.log("Program Ended");
```

### 28.repl1.js

```
const repl = require('repl');
const msg = 2+2;

repl.start('> ').context.m = msg;
```

### 29.s.js

```
var http = require('http');
var fs = require('fs');
var url = require('url');

http.createServer( function (request, response) {
   fs.readFile("index.html", function (err, data) {
      if (err) {
         console.log(err);

         response.writeHead(404, {'Content-Type': 'text/html'});
      } else {
```

```
         response.writeHead(200, {'Content-Type': 'text/html'});
         response.write(data.toString());
      }
            response.end();
   });
}).listen(8081);
console.log('Server running at http://127.0.0.1:8081/');
```

### 30.server.js

```
var express = require('express');
var app = express();
var fs = require("fs");
app.post('/listUsers', function (req, res) {
   fs.readFile( __dirname + "/" + "users.json", 'utf8', function (err, data) {
      console.log( data );
      res.end( data );
   });
})
var server = app.listen(8081, function () {
   var host = server.address().address
   var port = server.address().port
   console.log("Example app listening at http://%s:%s", host, port)
})
```

### 31.server1.js

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

var user = {
   "user4" : {
      "name" : "mohit",
      "password" : "password4",
      "profession" : "teacher",
      "id": 4
   }
}

app.post('/addUser', function (req, res) {
   // First read existing users.
   fs.readFile( __dirname + "/" + "users.json", 'utf8', function (err, data) {
      data = JSON.parse( data );
```

```
      data["user4"] = user["user4"];
      console.log( data );
      res.end( JSON.stringify(data));
   });
})

var server = app.listen(8081, function () {
   var host = server.address().address
   var port = server.address().port
   console.log("Example app listening at http://%s:%s", host, port)
})
```

### 32.server2.js

```
var express = require('express');
var MongoClient = require('mongodb').MongoClient;
var app = express();
var fs = require("fs");
var url = "mongodb://localhost:27017/";

app.get('/listUsers', function (req, res) {
   fs.readFile( __dirname + "/" + "users.json", 'utf8', function (err, data) {
      console.log( data );
      res.end( data );
   });
})

app.get('/adddata', function (req, res) {
MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("vcet");
  var myobj = { name: "VCET", address: "KT marg Vasai" };
  dbo.collection("inft").insertOne(myobj, function(err, res) {
    if (err) throw err;
    console.log("1 document inserted");
    db.close();
  });
});

app.post('/writedata', function (req, res) {
console.log("called");
const queryObject = url.parse(req.url,true).query;
MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("vcet");
  var myobj = queryObject;
  dbo.collection("inft").insertOne(queryObject, function(err, res) {
```

```
    if (err) throw err;
    console.log("1 document inserted");
    db.close();
  });
});
```

```
      res.end( "1 document inserted" );
   });
})
```

```
var server = app.listen(8081, function () {
   var host = server.address().address
   var port = server.address().port
   console.log("Example app listening at http://%s:%s", host, port)
})
```

### 33.server3.js

```
var express = require('express');
var MongoClient = require('mongodb').MongoClient;
var app = express();
var fs = require("fs");
var url = "mongodb://localhost:27017/";

app.get('/listUsers', function (req, res) {
   fs.readFile( __dirname + "/" + "users.json", 'utf8', function (err, data) {
      console.log( data );
      res.end( data );
   });
})

app.get('/adddata', function (req, res) {
MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("vcet");
  var myobj = { name: "VCET", address: "KT marg Vasai" };
  dbo.collection("inft").insertOne(myobj, function(err, res) {
    if (err) throw err;
    console.log("1 document inserted");
    db.close();
  });
```

```
});

app.get('/trydata', function (req, res) {
console.log("called");
//const queryObject = url.parse(req.url,true).query;
MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("vcet");
  var myobj = { name: "chandan", address: "Ratnagiri" };;
  dbo.collection("inft").insertOne(queryObject, function(err, res) {
    if (err) throw err;
    console.log("1 document inserted");
    db.close();
  });
//});

      res.end( "1 document inserted" );
   });
})



var server = app.listen(8081, function () {
   var host = server.address().address
   var port = server.address().port
   console.log("Example app listening at http://%s:%s", host, port)
})
```

## 34.server4.js

```
var net=require ("net");
var server = net.createServer();
server.on('connection', handleConnection);
server.listen(9000, function() {
  console.log('server listening to %j', server.address());
});
function handleConnection(conn) {
  var remoteAddress = conn.remoteAddress + ':' + conn.remotePort;
  console.log('new client connection from %s', remoteAddress);
 conn.on('data', onConnData);
 function onConnData(d) {
    console.log('connection data from %s: %j', remoteAddress, d);
    conn.write(d);
  }
}
```

### 35.simple.js

```
var http=require("http");
http.createServer(function(request,response)
{
response.writeHead(200,{'Content-Type':'text/plain'});
response.end("Hello VCET");
}).listen(8081);

console.log("server running");
```

### 36.simpleapp.js

```
var http=require("http");
http.createServer(function (request, response) {

   response.writeHead(200, {'Content-Type': 'text/plain'});

      response.end('Hello World\n');
}).listen(8081);

console.log('Server running at http://127.0.0.1:8081/');
```

### 37.sqlconnect.js

```
const sql = require('mssql');
const sqlConfig = {
  user: 'sa',
  password: 'chandan',
  database: 'vcet',
  port:55360,
  server: '127.0.0.1',

 pool: {
    max: 10,
    min: 0,
    idleTimeoutMillis: 30000
  },
  options: {
    encrypt: false, // for azure
    trustServerCertificate: false // change to true for local dev / self-
signed certs
  }
```

```
}



var a= async () => {
console.log("hello");
 try {
     var q=await sql.connect(sqlConfig);
   console.log(q);
    });

const result =await  sql.query("select * from inft");
console.log(result.recordsets);



   } catch (err) {
  console.log(err);
 }
}

a();
```

### 38.test1.js

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

http.createServer(function (req, res) {
  res.writeHead(400, {'Content-Type': 'text/html'});
  res.end('Hello World!');
}).listen(8080);
```

### 39.test2.js

```
var events = require('events');
var eventEmitter = new events.EventEmitter();
var eventEmitter1 = new events.EventEmitter();
var eventEmitter2 = new events.EventEmitter();

var myEventHandler = function () {
eventEmitter1.emit('scream1');
}
```

```
var myEventHandler1 = function () {
eventEmitter2.emit('scream2');
}

var myEventHandler2 = function () {
  console.log('I hear a scream!');
}



eventEmitter.on('scream', myEventHandler);
eventEmitter1.on('scream1', myEventHandler1);
eventEmitter2.on('scream2', myEventHandler2);


eventEmitter.emit('scream');
```

### 40.test3.js

```
var http = require('http');
var formidable = require('formidable');

http.createServer(function (req, res) {
    res.writeHead(200, {'Content-Type': 'text/html'});
    res.write('<form action="fileupload" method="post"
enctype="multipart/form-data">');
    res.write('</form>');
    return res.end();

}).listen(8080);
```

### 41.unblocking.js

```
var fs = require("fs");

fs.readFile('software.txt', function (err, data) {
   if (err) return console.error(err);
   console.log(data.toString());
});

console.log("Program Ended");
```

### 42.webclient.js

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

var options = {
   host: 'localhost',
   port: '8081',
};

var callback = function(response) {
   var body = '';
   response.on('data', function(data) {
      body += data;
   });

   response.on('end', function() {
      console.log(body);
   });
}
var req = http.request(options, callback);
req.end();
```

### 43. writerstream.js

```
var fs = require("fs");
var data = 'Simply Easy Learning';

// Create a writable stream
var writerStream = fs.createWriteStream('output.txt');

// Write the data to stream with encoding to be utf8
writerStream.write(data,'UTF8');

// Mark the end of file
writerStream.end();

// Handle stream events --> finish, and error
writerStream.on('finish', function() {
   console.log("Write completed.");
});

writerStream.on('error', function(err) {
   console.log(err.stack);
});

console.log("Program Ended");
```

### 44.google-services.json

```
{
  "project_info": {
    "project_number": "475602054971",
    "project_id": "test-dad7d",
    "storage_bucket": "test-dad7d.appspot.com"
  },
  "client": [
    {
      "client_info": {
        "mobilesdk_app_id": "1:475602054971:android:0bd383b31ccfbb8fdf911e",
        "android_client_info": {
          "package_name": "com.example.myapplication"
        }
      },
      "oauth_client": [
        {
          "client_id": "475602054971-
di04vnn61c64lev69ci65561m3vuni9k.apps.googleusercontent.com",
          "client_type": 3
        }
      ],
      "api_key": [
        {
          "current_key": "AIzaSyDC3a_zvpTnQktg2PC3smkdQ7ZmD1VRtmA"
        }
      ],
      "services": {
        "appinvite_service": {
          "other_platform_oauth_client": [
            {
              "client_id": "475602054971-
di04vnn61c64lev69ci65561m3vuni9k.apps.googleusercontent.com",
              "client_type": 3
            }
          ]
        }
      }
    }
  ],
  "configuration_version": "1"
}
```

### 45.index.html

```
<html>
   <head>
      <title>Sample Page</title>
   </head>

   <body>
      Hello World!
   </body>
</html>
```

## 46.mongofinal.txt

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("vcet");
  var myobj = { name: "VCET", address: "KT marg Vasai" };
  dbo.collection("inft").insertOne(myobj, function(err, res) {
    if (err) throw err;
    console.log("1 document inserted");
    db.close();
  });
});
```

## 47.output.txt

```
Simply Easy Learning
```

## 48.server3.txt

```
var express = require('express');
var MongoClient = require('mongodb').MongoClient;
var app = express();
var fs = require("fs");
var url = "mongodb://localhost:27017/";

app.get('/listUsers', function (req, res) {
   fs.readFile( __dirname + "/" + "users.json", 'utf8', function (err, data) {
      console.log( data );
      res.end( data );
   });
})

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

```
MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("vcet");
  var myobj = { name: "VCET", address: "KT marg Vasai" };
  dbo.collection("inft").insertOne(myobj, function(err, res) {
    if (err) throw err;
    console.log("1 document inserted");
    db.close();
  });
});

app.post('/writedata', function (req, res) {
console.log("called");
const queryObject = url.parse(req.url,true).query;
MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("vcet");
  var myobj = queryObject;
  dbo.collection("inft").insertOne(queryObject, function(err, res) {
    if (err) throw err;
    console.log("1 document inserted");
    db.close();
  });
});




     res.end( "1 document inserted" );
   });
})



var server = app.listen(8081, function () {
   var host = server.address().address
   var port = server.address().port
   console.log("Example app listening at http://%s:%s", host, port)
})
```

## 49.software.txt

```
192.168.10.137

admin
admin1234
```

## 50.test-dad7d-firebase-adminsdk-sz83s-bd946a5f85.json


```
{
  "type": "service_account",
  "project_id": "test-dad7d",
  "private_key_id": "bd946a5f851f3d07163354b1aeb3be85d8ccf628",
  "private_key": "-----BEGIN PRIVATE KEY-----
\nMIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQC+E5WXVG5IUhif\n/OhjEbUvpD
P7TSHj+Fk7LwdZsUO8dfcHZd8EosoHbSHLuZ1CuyrB1s9YpGFhVYDh\nAv1cnPo3mP22guT9Dv5/Pu
2Wbv3yo3oF15Bup31BCVbgj+RhwgKdrxUuvO+fijVO\nw4Luioo+J11yBqrv/t9ryLw1eQ5h+nep8w
O5PHZZ1z4IfO1mIk9u8djvkYuoEgHv\nSWGuCJdLxofrK4wMpyvHb3Xpzwn6Wp8ldg7zzBTsQ8WhQq
QIXAQROpaj+ni5IrFh\noCfAizZ/A7eAS8szxV7Qr5x9V1pnGEznRq3fLu6DvDnBVZrcEKzsuAqzIu
lDd22K\nw4j3XI9HAgMBAAECggEAFTPdD3TpGmMiARK2n3p1shtIeyNXAPWOipYL5VrrOeYU\nyLv5
3SwFij+HshCB9NQpCFdcaMBpZnChwDfTqFWQacxEnOvL/FiEVhBgc1PtOvg2\nX9l0cIt5YtxWVxVl
/euk3A8QN1b3Cd5iZOdVPyOwGQ/OEi6hOcU+Celp6TMYQiB7\nVTiJcIkuV8eEzhhRHs9Lrjs6ZtWg
XadBrdAbvTfJmrpSUXhdedJBY+1JVD1B8H52\nP9vUweq9fnBd9Hd4nGVyDR8ifWlsNtoPBGwEZXUu
nIdVbBFXTpjsEdtTZxsOXyb+\nOny3H4h5BVJALHnjWLh9rDvn4bIcuawCmmFBMEL1/QKBgQD35oKY
LxNLXITHTO/Z\nkJRYvEmAqlCV8pXCAdBWORL6m/4htmtRycov8w1XnVOgP85CybopCXbOv8NZw0KT
\nB7eNr3HKhaVLNnpzvvmYeSP+dyaxeatnQUZeubXubbmit/IvonqnwweZiQQkUiFB\ndM2A1REIln
/pmuBSLq2EV9lhrQKBgQDESWxMwZcckD7ZxwPnaMYQIzn/f1SQHI6R\n8AlewOmtVlTj7azY0RTLkX
mWy5PL/44RcX6zKlioD4vpOGtt5InVPmRciBcsfDjI\nXD7XpQ5yDGH0PXkjtWqdK9mKrXYzSnWBLR
d3b/vR7maUR5ROGzDy89N6Kv6OauFM\n93viJa3bQwKBgD5G2mgA64fzJigrW9X8TEg+hPid0MDawG
SMsBn5HJt4tM8jVLKn\nTvPcy21HSMrpPJfqtam5hu9JnXnfY8osFiCyApQuHuhWN0z8nReA11oF9O
KA/l5i\nN5t0bBAJ4KjQLTdqSgX7KN2ZXkEBoPkgrr++xxixNlEpZ1E+CLlGmIXBAoGAYsq1\nmHhD
LxwmlgOFon3oWgzwzvhuPk4xL6CJhsg/lH+EIGfqfAHXnpGsicoSrDUGcx8V\nbuWcU5HnkE1WPKwu
90X8y1mvmDM5fRZ4hp7Lg3lZ/8zE4MUZUNvWFOczO6N5OrfQ\nPMv/CW0c8fqI2Tr0SwOPZ8il8FY4
6T0SfAMM2vMCgYAkSV0aF5LnmqVwieLd5GrS\njHGgtfLTUtSgh70AHlp9xtQJ7dcetZecKeajQC6j
z4pmGUYwhERlBZuiTqo2suQP\nAyp0AhRK3dm/Xzo4dOFzbPP1+wAd2Nj2TximdqC8UF+9yUa0pJbb
DqVAR+bO4m5g\nf2sEuLm8BGH48nMPJkn4bA==\n-----END PRIVATE KEY-----\n",
  "client_email": "firebase-adminsdk-sz83s@test-
dad7d.iam.gserviceaccount.com",
  "client_id": "113192907082711476351",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://oauth2.googleapis.com/token",
  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
  "client_x509_cert_url":
"https://www.googleapis.com/robot/v1/metadata/x509/firebase-adminsdk-
sz83s%40test-dad7d.iam.gserviceaccount.com"
}
```


## 51.users.json


```
{
   "user1" : {
      "name" : "mahesh",
```

```
      "password" : "password1",
      "profession" : "teacher",
      "id": 1
   },

   "user2" : {
      "name" : "suresh",
      "password" : "password2",
      "profession" : "librarian",
      "id": 2
   },

   "user3" : {
      "name" : "ramesh",
      "password" : "password3",
      "profession" : "clerk",
      "id": 3
   }
}
]
```

# inside module 5

1. **block.js**

```
const fs = require('fs');
const data = fs.readFileSync('mongo.js'); // blocks here until file is read
console.log(data);
console.log("hello");
```

2. **sample.js**

```
var http = require("http");

http.createServer(function (request, response) {
   response.writeHead(200, {'Content-Type': 'text/plain'});
   response.end('Hello World\n');
}).listen(8081);

console.log('Server running at http://127.0.0.1:8081/');
```

# REACT

### 3. App.js

```
import logo from './logo.svg';
import './App.css';
import Hook2 from './Hook2.js';
function App() {
var chandan=""
  return (
      <div>
<p> lets check router </p>
<Hook2></Hook2>
      </div>
  );
}

export default App;
```

### 4. App.test.js

```
import { render, screen } from '@testing-library/react';
import App from './App';

test('renders learn react link', () => {
  render(<App />);
  const linkElement = screen.getByText(/learn react/i);
  expect(linkElement).toBeInTheDocument();
});
```

### 5. Example.js

```
import React, { useState,useEffect } from 'react';

function Example() {
  // Declare a new state variable, which we'll call "count"
  const [count, setCount] = useState(0);
useEffect(() => {
    // Update the document title using the browser API
    document.title = `You clicked ${count} times`;
  });
```

```
  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>
        Click me
      </button>
    </div>
  );
}

export default Example;
```

6. Example1.js

```
import React, {useState,useEffect} from 'react';

function useMyCustomHook(data) {


    useEffect(() => {
document.title=data;
    }, [data]);

    return 5;
}

function MyAwesomeComponent({someDataKey}) {

    const someValue = useMyCustomHook("vcet");

    return (<p>The new value is {someValue}</p>);
}

export default Example1;
```

7. Example2.js

```
import React, {useState,useEffect} from 'react';

class Example2 extends React.Component {
  constructor(props) {
    super(props)

    // Create the ref
    this.exampleRef = React.createRef()
```

```
  }

  render() {
    return (
      <div>
        // Call the ref with the `ref` attribute
        <input type="text" ref={this.exampleRef} onChange={this.handleref} />
      </div>
    );
  }
handleref(event){
alert("Hello! I am an alert box!!");
}
}

export default Example2;
```

8. Hello.js

```
import React, { Component } from 'react';

class Button extends Component {
  constructor(props) {
    super(props);
    this.state = {
      isDisplayed: true,
    };
  }
  toggleShowHide = () => {
    this.setState(state => ({ isDisplayed: !state.isDisplayed }));
  };
  render() {
    return (
      <button onClick={this.toggleShowHide} type="button">
        Toggle Show/Hide
      </button>
    );
  }
}



class Hello extends Component {
  render() {
    const hello = 'Say Hello to learning Props/State in React!';
    return (
      <div>
```

```
  {Button.isDisplayed ? <HelloReact hello={hello} /> : null}
        <Button />
      </div>
    );
  }
}
const HelloReact = ({ hello }) => <h1>{hello}</h1>;



export default Hello;
```

9. hook1.js

```
import React, { useCallback } from 'react';
import React, { useState, useCallback } from 'react'
const Hello1 = () => {
const [count, setCount] = useState(0)
const incrementCounter = useCallback(() => {
setCount(count + 1)
}, [count])
const decrementCounter = useCallback(() => {
setCount(count - 1)
}, [count])
return (
    <div>
    <h3>Count: {count}</h3>
    <button onClick={incrementCounter}>Increase counter</button>
    <br />
    <button onClick={decrementCounter}>Decrease Counter</button>
    </div>
)
}
export default Example2;export default Hook1;
```

10. hook2.js

```
import React, {useState,useEffect} from 'react';
function useMyCustomHook(data) {

    useEffect(() => {
document.title=data;
    }, [data]);
    return data;
}
```

```
function Hook2() {
    const someValue = useMyCustomHook("vcet1");
    return (<p>The new value is {someValue}</p>);
}

export default Hook2;
```

11. Hook3.js

```
import React,{useEffect,useState} from 'react';

class Example3 extends React.Component {
  constructor(props) {
    super(props)

    // Create the ref
    this.exampleRef = React.createRef()
  }

  render() {
    return (
      <div>
        // Call the ref with the `ref` attribute
        <input type="text" ref={this.exampleRef} />
      </div>
    )
  }
}



export default Example3
```

12. Index.css

```
body {
  margin: 0;
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto',
'Oxygen',
    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
    sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}
```

```
code {
  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
    monospace;
}
```

13. Index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById('root')
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

14. logo.svg

```
<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 841.9 595.3"><g
fill="#61DAFB"><path d="M666.3 296.5c0-32.5-40.7-63.3-103.1-82.4 14.4-63.6 8-
114.2-20.2-130.4-6.5-3.8-14.1-5.6-22.4-5.6v22.3c4.6 0 8.3.9 11.4 2.6 13.6 7.8
19.5 37.5 14.9 75.7-1.1 9.4-2.9 19.3-5.1 29.4-19.6-4.8-41-8.5-63.5-10.9-13.5-
18.5-27.5-35.3-41.6-50 32.6-30.3 63.2-46.9 84-46.9V78c-27.5 0-63.5 19.6-99.9
53.6-36.4-33.8-72.4-53.2-99.9-53.2v22.3c20.7 0 51.4 16.5 84 46.6-14 14.7-28
31.4-41.3 49.9-22.6 2.4-44 6.1-63.6 11-2.3-10-4-19.7-5.2-29-4.7-38.2 1.1-67.9
14.6-75.8 3-1.8 6.9-2.6 11.5-2.6V78.5c-8.4 0-16 1.8-22.6 5.6-28.1 16.2-34.4
66.7-19.9 130.1-62.2 19.2-102.7 49.9-102.7 82.3 0 32.5 40.7 63.3 103.1 82.4-
14.4 63.6-8 114.2 20.2 130.4 6.5 3.8 14.1 5.6 22.5 5.6 27.5 0 63.5-19.6 99.9-
53.6 36.4 33.8 72.4 53.2 99.9 53.2 8.4 0 16-1.8 22.6-5.6 28.1-16.2 34.4-66.7
19.9-130.1 62-19.1 102.5-49.9 102.5-82.3zm-130.2-66.7c-3.7 12.9-8.3 26.2-13.5
39.5-4.1-8-8.4-16-13.1-24-4.6-8-9.5-15.8-14.4-23.4 14.2 2.1 27.9 4.7 41 7.9zm-
45.8 106.5c-7.8 13.5-15.8 26.3-24.1 38.2-14.9 1.3-30 2-45.2 2-15.1 0-30.2-.7-
45-1.9-8.3-11.9-16.4-24.6-24.2-38-7.6-13.1-14.5-26.4-20.8-39.8 6.2-13.4 13.2-
```

```
26.8 20.7-39.9 7.8-13.5 15.8-26.3 24.1-38.2 14.9-1.3 30-2 45.2-2 15.1 0 30.2.7
45 1.9 8.3 11.9 16.4 24.6 24.2 38 7.6 13.1 14.5 26.4 20.8 39.8-6.3 13.4-13.2
26.8-20.7 39.9zm32.3-13c5.4 13.4 10 26.8 13.8 39.8-13.1 3.2-26.9 5.9-41.2 8
4.9-7.7 9.8-15.6 14.4-23.7 4.6-8 8.9-16.1 13-24.1zM421.2 430c-9.3-9.6-18.6-
20.3-27.8-32 9 .4 18.2.7 27.5.7 9.4 0 18.7-.2 27.8-.7-9 11.7-18.3 22.4-27.5
32zm-74.4-58.9c-14.2-2.1-27.9-4.7-41-7.9 3.7-12.9 8.3-26.2 13.5-39.5 4.1 8 8.4
16 13.1 24 4.7 8 9.5 15.8 14.4 23.4zM420.7 163c9.3 9.6 18.6 20.3 27.8 32-9-.4-
18.2-.7-27.5-.7-9.4 0-18.7.2-27.8.7 9-11.7 18.3-22.4 27.5-32zm-74 58.9c-4.9
7.7-9.8 15.6-14.4 23.7-4.6 8-8.9 16-13 24-5.4-13.4-10-26.8-13.8-39.8 13.1-3.1
26.9-5.8 41.2-7.9zm-90.5 125.2c-35.4-15.1-58.3-34.9-58.3-50.6 0-15.7 22.9-35.6
58.3-50.6 8.6-3.7 18-7 27.7-10.1 5.7 19.6 13.2 40 22.5 60.9-9.2 20.8-16.6
41.1-22.2 60.6-9.9-3.1-19.3-6.5-28-10.2zM310 490c-13.6-7.8-19.5-37.5-14.9-75.7
1.1-9.4 2.9-19.3 5.1-29.4 19.6 4.8 41 8.5 63.5 10.9 13.5 18.5 27.5 35.3 41.6
50-32.6 30.3-63.2 46.9-84 46.9-4.5-.1-8.3-1-11.3-2.7zm237.2-76.2c4.7 38.2-1.1
67.9-14.6 75.8-3 1.8-6.9 2.6-11.5 2.6-20.7 0-51.4-16.5-84-46.6 14-14.7 28-31.4
41.3-49.9 22.6-2.4 44-6.1 63.6-11 2.3 10.1 4.1 19.8 5.2 29.1zm38.5-66.7c-8.6
3.7-18 7-27.7 10.1-5.7-19.6-13.2-40-22.5-60.9 9.2-20.8 16.6-41.1 22.2-60.6 9.9
3.1 19.3 6.5 28.1 10.2 35.4 15.1 58.3 34.9 58.3 50.6-.1 15.7-23 35.6-58.4
50.6zM320.8 78.4z"/><circle cx="420.9" cy="296.5" r="45.7"/><path d="M520.5
78.1z"/></g></svg>
```

15. NameForm.js

```
import React from 'react';
class NameForm extends React.Component {
  constructor(props) {
    super(props);
    this.state = {value: ''};

    this.handleChange = this.handleChange.bind(this);
    this.handleSubmit = this.handleSubmit.bind(this);
  }

  handleChange(event) {
 React.createElement("h1", null, "Baked Salmon")
    this.setState({value: event.target.value});
  }

  handleSubmit(event) {
React.createElement("h1", null, "Baked Salmon")
alert('A name was submitted: ' + this.state.value);
    event.preventDefault();
  }

  render() {
    return (
      <form onSubmit={this.handleSubmit}>
<p id='rc'>This is test</p>
```

```
        <label>
          Name:
          <input type="text" value={this.state.value}
onChange={this.handleChange} />
        </label>
        <input type="submit" value="Submit" />
{
React.createElement("ul", {"className": "ingredients"},
React.createElement("li", null, "1 lb Salmon"),
React.createElement("li", null, "1 cup Pine Nuts"),
React.createElement("li", null, "2 cups Butter Lettuce"),
React.createElement("li", null, "1 Yellow Squash"),
React.createElement("li", null, "1/2 cup Olive Oil"),
React.createElement("li", null, "3 cloves of Garlic")
)
}


      </form>
    );
  }
}

export default NameForm;
```

16. NameForm1.js

```
import React from 'react';

class NameForm1 extends React.Component{
constructor(Props)
{super(Props);
this.state={value:''};
}

render()
{
return(
<form onSubmit={this.handleSubmit.bind(this)}>
<label> Name is
<input type="text" value={this.state.value} />
</label>
<input type ="Submit" value="Submit"/>
</form>
);
}
```

```
handleSubmit(event)
{
alert("name is" + this.state.value);
event.preventDefault();
}
}

export default NameForm1;
```

17. New Text Document.txt

```
varx=5,y=1
var obj ={ x:10}
with(obj)
{
    alert(y)
}
```

18. Numberlist.js

```
import React from 'react';

function NumberList(Props)
{
const numbers = [1, 2, 3, 4, 5];
return(  <number numbers={Props} />);
}

export default NumberList;
```

19. reportWebVitals.js

```
const reportWebVitals = onPerfEntry => {
  if (onPerfEntry && onPerfEntry instanceof Function) {
    import('web-vitals').then(({ getCLS, getFID, getFCP, getLCP, getTTFB }) =>
{
      getCLS(onPerfEntry);
      getFID(onPerfEntry);
      getFCP(onPerfEntry);
      getLCP(onPerfEntry);
      getTTFB(onPerfEntry);
    });
  }
```

```
};

export default reportWebVitals;
```

20. setupTests.js

```
// jest-dom adds custom jest matchers for asserting on DOM nodes.
// allows you to do things like:
// expect(element).toHaveTextContent(/react/i)
// learn more: https://github.com/testing-library/jest-dom
import '@testing-library/jest-dom';
```

21. test1.js

```
import React from 'react';
import ReactDOM from 'react-dom';

const Demo=()=>
{
  return <h1>Welcome to vcet</h1>;
}

export default Demo;
```

22. test2.js

```
import React from 'react';

const Welcome=()=>
{
return (
    <h1>Welcome to vcet</h1>
    );
}

const functionExample=()=>
{
    return (
    <Welcome/>
    );
}
```

```
export default Welcome;
```

23. test3.js

```
import React, { useEffect } from 'react';

const Example=()=> {
  useEffect(() => {
    console.log("Mounting...");
  });
      return (
      <h1>
        vcet....!
      </h1>
      );
}

export default Example;
```

24. Trial.js

```
import React from 'react';

class Trial extends React.Component{
state={"college":"vcet","address":"vasai"};
render(){
return(
    <div>
    <p>This is monday morning</p>
    <p>have a nice day</p>
    <p>{this.defaultProps.name }</p>
    <p>{this.defaultProps.surname }</p>
    <p>{this.state.college}</p>
    <p>{this.state.address}</p>
    <p>{Date()}</p>
    <p> {this.f("vcet")}</p>
    </div>
    );
}

f(name)
{
let a= new abc("vidyavardhini");
return(a.display());
}
```

```
defaultProps=
{
name:"chandan",
surname:"kolvankar"
}
}


class abc{
constructor(Props)
    { this.state={"college":Props.name}}

display()
    {
    return(this.state.college);
    }

}

export default Trial;
```

25. Trial2.js

```
import React from 'react';

class Trial2 extends React.Component{
constructor(Props)
{
super();
this.state={"name":Props.name};}

render(){
return(
    <div>
    <p>This is monday morning</p>
    <p>{this.state.name}</p>
    </div>
    );
this.forceUpdate(()=>{this.setState({name:"vcet"});});

}

componentDidUpdate() {


this.setState({name:"patil"});
```

```
}



}

export default Trial2;
```

26. try.js

```
import React from 'react';
import './App.css';
class Try extends React.Component{
state={"companyName":""};
render(){
return(
    <div>
                <h2>Simple Event Example</h2>
                <label htmlFor="name">Enter company name: </label>
                <input type="text" id="companyName"
onChange={this.changeText.bind(this)}/>
                <h4>You entered: { this.state.companyName }</h4>
            <button id="companyName" onClick={() =>
this.handler.bind(this,{"name":"vidyavardhini"})}>
          Click Here
        </button>
<h4>You entered: { this.state.companyName }</h4>

</div>
);
}

 handler = (name) => {
    // Changing the state
    this.setState({ companyName: name });
  };

 changeText(event) {
        this.setState({
            companyName: event.target.value
        });
    }
}

export default Try;
```