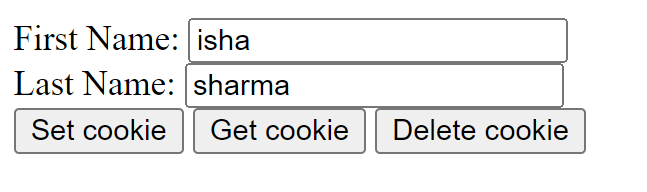
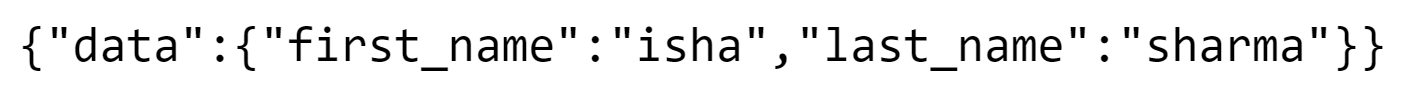
CA-2 (CODE BASED TEST)

SET A

Q1. CREATE THE FOLLOWING SCENERIO WHERE YOU CAN SET, DELETE AND VIEW COOKIE VALUE

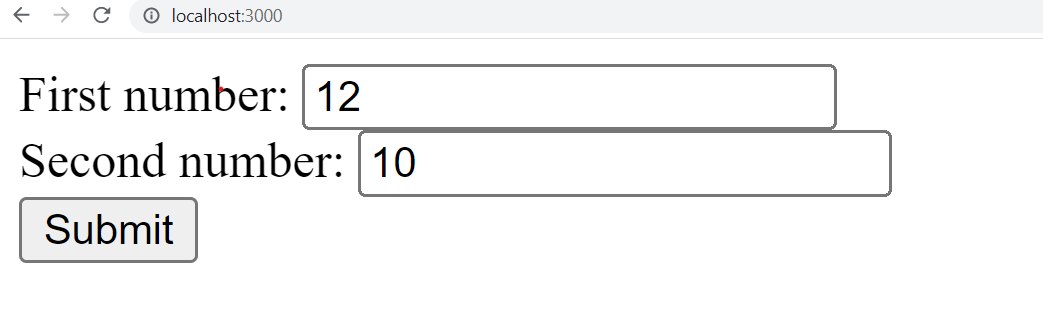


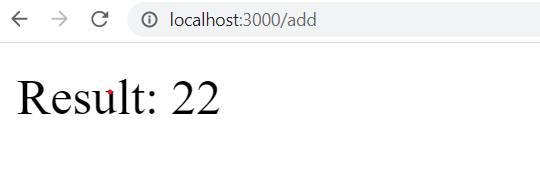


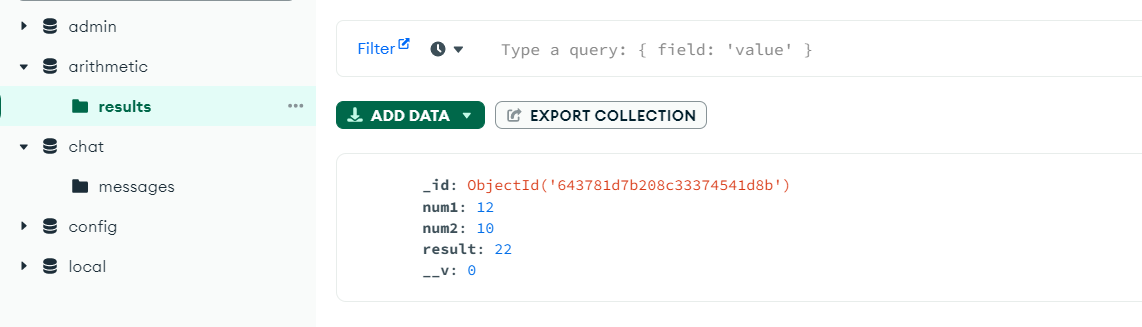




Q2 take two input numbers from user perform sum and save the data in database and display the answer

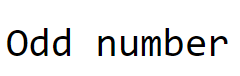




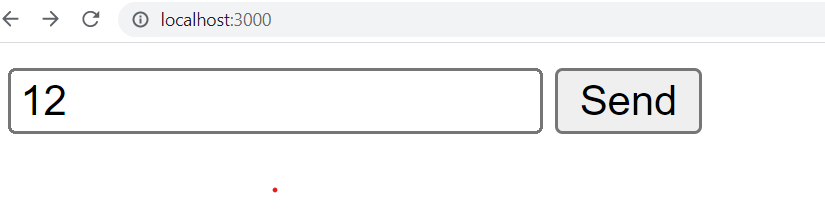


Q1.using sockets find the number entered is odd and even and save the result in “test.txt” file.

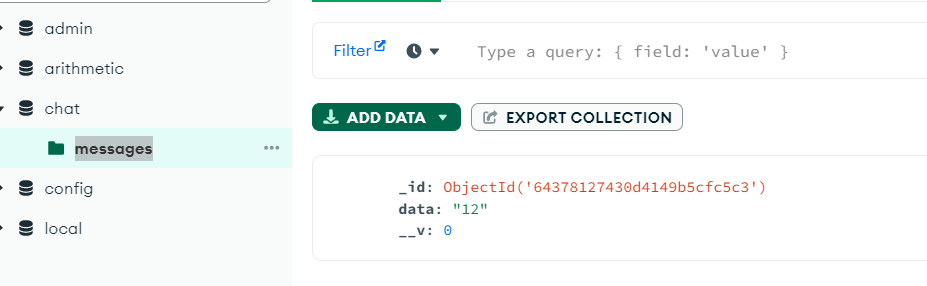




Q2.Insert data from user in textbox using socket and then save that data in database







ANSWERS:

SETAQ1.HTML:

<html>

<body>

<form method="POST">

First Name: <input type="text" name="first\_name">  <br>

Last Name: <input type="text" name="last\_name">  <br>

<button type="submit" formaction="/set">Set cookie</button>

<button type="submit" formaction="/get">Get cookie</button>

<button type="submit" formaction="/delete">Delete cookie</button>

</form>

</body>

</html>

SETAQ1.JS

var express = require('express');

var app = express();

var cookieParser = require('cookie-parser');

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

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

app.use(cookieParser());

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

  res.sendFile( \_\_dirname + "/" + "setaq1.html" );

})

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

  res.cookie("data", req.body);

  res.send("cookie set");

})

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

  console.log(req.cookies);

  res.send(req.cookies);

})

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

  res.clearCookie("data");

 res.send("Cookie has been deleted");

})

app.listen(3000);

SETAQ2.HTML

<html>

<body>

<form action="/add" method="POST">

First number: <input type="text" name="num1">  <br>

Second number: <input type="text" name="num2">  <br>

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

</form>

</body>

</html>

SETAQ2.JS

const express = require('express');

const mongoose = require('mongoose');

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

const app = express();

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

// connect to MongoDB

mongoose.connect('mongodb://0.0.0.0/arithmetic', { useNewUrlParser: true })

  .then(() => console.log('MongoDB Connected'))

  .catch(err => console.log(err));

// create a schema for the result model

const resultSchema = new mongoose.Schema({

  num1: Number,

  num2: Number,

  result: Number

});

// create a model for the result

const Result = mongoose.model('Result', resultSchema);

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

   res.sendFile( \_\_dirname + "/" + "SetAQ2.html" );

});

// set up a route to handle the form submission

app.post('/add', (req, res) => {

  const num1 = parseInt(req.body.num1);

  const num2 = parseInt(req.body.num2);

  const result = num1 + num2;

  // create a new result object

  const newResult = new Result({

    num1: num1,

    num2: num2,

    result: result

  });

  // save the new result to the database

  newResult.save()

    .then(() => {

      res.send(`Result: ${result}`);

    })

    .catch(err => {

      console.log(err);

      res.send('Error saving result to database');

    });

});

// start the server

app.listen(3000, () => console.log('Server started on port 3000'));

SETBQ1.HTML

<!DOCTYPE html>

<html>

   <head>

      <title>program</title>

   </head>

   <body>

      <form method="POST" action="/insert-data">

      <input type="text" id="data", name="data">

      <input type="submit" value="Send" onclick="a()">

   </form>

      <script src = "/socket.io/socket.io.js"></script>

      <script>

         var socket = io();

   function a(){

     socket.emit('clientEvent', document.getElementById('data').value);

     }

      </script>

   </body>

</html>

SETBQ1.JS

const express = require('express');

const mongoose = require('mongoose');

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

const app = express();

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

// connect to MongoDB

mongoose.connect('mongodb://0.0.0.0/arithmetic', { useNewUrlParser: true })

  .then(() => console.log('MongoDB Connected'))

  .catch(err => console.log(err));

// create a schema for the result model

const resultSchema = new mongoose.Schema({

  num1: Number,

  num2: Number,

  result: Number

});

// create a model for the result

const Result = mongoose.model('Result', resultSchema);

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

   res.sendFile( \_\_dirname + "/" + "SetAQ2.html" );

});

// set up a route to handle the form submission

app.post('/add', (req, res) => {

  const num1 = parseInt(req.body.num1);

  const num2 = parseInt(req.body.num2);

  const result = num1 + num2;

  // create a new result object

  const newResult = new Result({

    num1: num1,

    num2: num2,

    result: result

  });

  // save the new result to the database

  newResult.save()

    .then(() => {

      res.send(`Result: ${result}`);

    })

    .catch(err => {

      console.log(err);

      res.send('Error saving result to database');

    });

});

// start the server

app.listen(3000, () => console.log('Server started on port 3000'));

SETBQ2.HTML

<!DOCTYPE html>

<html>

   <head>

      <title>program</title>

   </head>

   <body>

      <form method="POST" action="/insert-data">

      <input type="text" id="data", name="data">

      <input type="submit" value="Send" onclick="a()">

   </form>

      <script src = "/socket.io/socket.io.js"></script>

      <script>

         var socket = io();

   function a(){

     socket.emit('clientEvent', document.getElementById('data').value);

     }

      </script>

   </body>

</html>

SETBQ2.JS

var express = require('express');

var app = express();

var http = require('http').createServer(app);

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

var io = require('socket.io')(http);

const mongoose = require('mongoose');

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

mongoose.connect('mongodb://0.0.0.0/chat', { useNewUrlParser: true })

  .then(() => console.log('MongoDB Connected'))

  .catch(err => console.log(err));

const msgSchema = new mongoose.Schema({

   data:String

   });

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

   res.sendFile( \_\_dirname + "/" + "setbq2.html" );

});

app.post('/insert-data', function(req, res) {

      const Msg = mongoose.model('messages', msgSchema)

      let data = new Msg(req.body)

      data.save()

      res.send("Message sent and saved to database successfully");

});

io.on('connection', function(socket) {

   socket.on('clientEvent', function(data) {

      console.log(data);

   });

});

http.listen(3000, function() {

   console.log('listening on localhost:3000');

});