

Dashboard / My units / FIT2095\_S2\_2021 / Week 2 / Week 2:Workshop Quiz

# Week 2: Workshop Quiz

# Week 2 Workshop Quiz

Q1)

For the URL provided below, Identify the following: (10m)

- 1. The query string
- 2. The protocol
- 3. The port number
- 4. The pathname
- 5. The server address

http://localhost:4000/weeks/2/wsq/?q=1&mark=20

### Q2)

Write a backend server in Node.js with the following specifications: (20m)

- 1. listens to pot 5678
- 2. responds with the string "Its S2 2021" for the pathname "/sem"
- 3. responds with the string "Workshop Quiz 2" for the pathname "/quiz"
- 4. responds with the string "Unknown pathname" for any other pathnames.

### Q3)

What commands/expressions do you have to use to perform the following? provide an example for each case. (9m)

- 1. Import an external package.
- 2. Create a new package.json file
- 3. Insert a new item to the list of dependencies in pakcage.json file

Last modified: Monday, 2 August 2021, 5:13 PM



<u>Dashboard</u> / My un	its / FIT2095_S2_2021 / Assessments / Week 2: Workshop Quiz
Started on	Monday, 2 August 2021, 8:50 PM
State	Finished
Completed on	Monday, 2 August 2021, 10:21 PM
Time taken	1 hour 31 mins
Grade	8.85 out of 10.00 (89%)
Print friendly format	
Question 1	
Complete	
Mark 8.85 out of 10.00	
a o.oo oat o. To.oo	
Question 1	
1. The query str	ing:
q=1&mark=20	
2. The protocol	
http://	
2. The part num	shor.
3. The port num	ibei .
4. The pathnam	
/weeks/2/wsq	/?
5. The server ac	ldress:
localhost:40	
Commont:	
Comment: Q1=95% Q2=95%	03=67%
Q1-35% Q2-35%	Q0-07 //
"localhost" is the ric	ght answer and not "localhost:4000" as 4000 is the port that you're listening on.
Please include enc	oding for every http response in the response.end().
Question 2	
Complete	
Not graded	

```
let http= require("http");
function handleRequest(request, response){
  let url=request.url;

  if(url==='/sem')
  response.end("lts S2 2021");
  else if(url==='/quiz')
  response.end("Workshop Quiz 2");
  else if(url==='/')
  response.end("Unknown pathname");

}
http.createServer(handleRequest).listen(5678);
```

Question **3**Complete

Not graded

#### Question 3

- 1. Import an external package.
  - o to import an external package, we have to use the 'require' command
- 2. Create a new package.json file
- To create a new package.json file run this command: npm init
  - 3.Insert a new item to the list of dependencies in pakcage.json file

Question **4**Not answered
Not graded

Question 5 Not sourcered  Question 5  Question 6 Not sourcered Not graded  Question 6  Not sourcered Not graded  Question 7  Not answered Not graded  Question 7			
Not answered Not graded  Question 6 Not answered Not graded  Question 6  Question 6  Not answered Not graded  Question 7 Not answered Not graded	Ouestion 5		
Question 5  Question 6 Not answered Not graded  Question 6  Question 7 Not answered Not graded  Question 7 Not answered Not graded			
Question 5  Question 6 Not answered Not gradied  Question 6			
Question 6 Not answered Not graded  Question 6  Question 7 Not answered Not graded	That graded		
Question 6 Not answered Not graded  Question 6  Question 7 Not answered Not graded			
Not answered Not answered Question 6  Question 7 Not answered Not graded	Question 5		
Not answered Not answered Question 6  Question 7 Not answered Not graded			
Not answered Not answered Question 6  Question 7 Not answered Not graded			
Not answered Not answered Question 6  Question 7 Not answered Not graded			
Not answered Not answered Question 6  Question 7 Not answered Not graded			
Not answered Not answered Question 6  Question 7 Not answered Not graded			
Not answered Not answered Question 6  Question 7 Not answered Not graded			
Not answered Not answered Question 6  Question 7 Not answered Not graded			
Not answered Not answered Question 6  Question 7 Not answered Not graded			
Not answered Not answered Question 6  Question 7 Not answered Not graded			
Not answered Not answered Question 6  Question 7 Not answered Not graded			
Not answered Not answered Question 6  Question 7 Not answered Not graded			
Not answered Not answered Question 6  Question 7 Not answered Not graded			
Not answered Not answered Question 6  Question 7 Not answered Not graded			
Not answered Not answered Question 6  Question 7 Not answered Not graded			
Not answered Not answered Question 6  Question 7 Not answered Not graded			
Not answered Not answered Question 6  Question 7 Not answered Not graded			
Not answered Not answered Question 6  Question 7 Not answered Not graded	Question 6		
Question 6  Question 7 Not answered Not graded			
Question 6  Question 7  Not answered Not graded			
Question 7 Not answered Not graded	Not graded		
Question 7 Not answered Not graded			
Question 7 Not answered Not graded	Question 6		
Not answered Not graded			
Not answered  Not graded			
Not answered  Not graded			
Not answered  Not graded			
Not answered  Not graded			
Not answered  Not graded			
Not answered  Not graded			
Not answered  Not graded			
Not answered  Not graded			
Not answered  Not graded			
Not answered  Not graded			
Not answered  Not graded			
Not answered  Not graded			
Not answered  Not graded			
Not answered Not graded			
Not answered  Not graded			
Not answered  Not graded	Outstien 7		
Not graded			
Ouestion 7	Not graded		
Ouestion 7			

Question 8	
Not answered	
Not graded	

### ■ Ed Discussion

Jump to...

Week 3: Workshop Quiz ▶



Dashboard / My units / FIT2095\_S2\_2021 / Week 3 / Week 3: Workshop Quiz

# Week 3: Workshop Quiz

# Q1) (12 marks)

Using Node.js and Express.js, develop a back end server that:

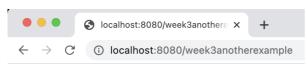
1. Accepts requests with the prefix 'week3' and responds with the welcome message 'Welcome to Week 3'.

Examples:

- http://localhost:8080/week3
- http://localhost:8080/week3labs
- http://localhost:8080/week3tutes
- http://localhost:8080/week3anotherexample



Welcome to week 3



Welcome to week 3

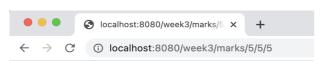
2. Accept requests with the pathname 'week3/marks'. The pathname must also contain three parameters that represent the three assessments in FIT2095, which are: pre-reading quiz (prerq), workshop quiz (wsq), and laboratory(lab). The handler must respond with the total mark of the week which is calculated by:

weekMark=prerq\*0.1+wsq\*0.1+lab\*0.2

Examples



Week 3 Mark is 4



Week 3 Mark is 2

# Q2)(2 marks)

Briefly explain two differences between route parameters and route query string. Support your answer with examples.

Last modified: Monday, 9 August 2021, 3:02 PM



Dashboard / My units / FIT2095\_S2\_2021 / Assessments / Week 3: Workshop Quiz

```
Started on Monday, 9 August 2021, 5:38 PM

State Finished

Completed on Monday, 9 August 2021, 8:15 PM

Time taken 2 hours 37 mins

Grade 5.00 out of 10.00 (50%)
```

### Print friendly format

```
Question 1
Complete
Mark 5.00 out of 10.00
```

```
Question 1
let express = require('express');
let app = express();
app.get('/week3', function (req, res) {
  console.log('Welcome to week 3');
  res.send('Welcome to week 3');
});
app.get('/week3labs', function (req, res) {
  console.log('Welcome to week 3');
  res.send('Welcome to week 3');
});
app.get('/week3tutues', function (req, res) {
  console.log('Welcome to week 3');
  res.send('Welcome to week 3');
app.get('/week3anotherexample', function (req, res) {
  console.log('Welcome to week 3');
  res.send('Welcome to week 3');
app.listen(8080);
let express = require('express');
let url = require('url');
let app = express();
let db = [];
let mk = {
  prerq: 10,
  awsq: 10,
  lab: 10
```

#### Ouestion 2

Not graded

Briefly explain two differences between route parameters and route query string, your answer with examples.

Route parameters are named URL segments that are used to capture the values specified at their position in the URL. For example lms.monash,edu/mod in this case params is 'mod' which leads us to moodle and it is our destination, and query string can be seen as some options we can use for example stuid=xxxx. or week=3 something like that

route params don't have actual value that leads us to what we seek for, it contains values within it, and route query string holds these values for us for example it holds the ID of a student or a ID for a book.

Question 3

Not answered

Not graded

Question <b>4</b> Not answered			
Not answered			
Not graded			
Question 4			

Question **5**Not answered

Not graded

Question 5

Question **6**Not answered

Not graded

Question 7	
Not answered	
Not graded	

Question **8**Not answered
Not graded

### ■ Week 2: Workshop Quiz

Jump to...

Week 4: Workshop Quiz ▶



```
<u>Dashboard</u> / My units / <u>FIT2095_S2_2021</u> / <u>Week 4</u> /
```

### Week4: Workshop Quiz

# Q1)

Given the following excerpt from a backend server, write a piece of code that is responsible for adding two key-value pairs for every new incoming request. The two pairs are:

unitCode:'FIT2095'

WeekNumber=4

PS: the backtick (``) characters are used to build a string template.

# Q2)

In the following piece of code, briefly explain the role of 'action' and 'method' attributes in the form tag.

```
<html>
<body>
<form action="/data" method="POST">

        User Name:
        <input type="text" name="username" /> </br>
        User Age:
        <input type="number" name="userage" /> </br>
        <button>Submit</button>
</form>
</body>
</html>
```

# Q3)

Rewrite app.js such that

- 1. Extracts value1 and value2 from the POST request that is sent by index.html on pathname='/findmax'
- 2. Responds back after rendering 'response.html'. The template shows the maximum value between value1 and value2

```
const express = require("express");
const app = express();
// your code goes here
app.use(function (req, res, next) {
    req.unitCode = "FIT2095";
   req.weekNumber = 4;
   next();
});
app.use(express.urlencoded({
    extended: true
}));
app.use(express.json());
//your code here
app.get("/", function (req, res) {
    res.sendFile(__dirname + "/index.html");
// your code here
app.listen(8080);
```

index.html

response.html

Last modified: Monday, 16 August 2021, 9:06 PM

■ VS Code keyboard-shortcuts-macos

```
Jump to...
```



Dashboard / My units / FIT2095 S2 2021 / Assessments / Week 4: Workshop Quiz

Started on Tuesday, 17 August 2021, 1:47 AM

State Finished

Completed on Tuesday, 17 August 2021, 2:30 AM

Time taken 43 mins 1 sec

Grade 7.20 out of 10.00 (72%)

### Print friendly format

Question 1

```
Question 1
Complete
Mark 7.20 out of 10.00
```

```
Question 2
Complete
Not graded
```

#### Question 2

Q1=100% Q2=100% Q3=36%

action: another name for pathname method: tells that a parse request is needed to be sent

```
Question 3
Complete
Not graded
```

```
const express = require("express");
const app = express();

let db=[];

app.use(function (req, res, next) {
    req.unitCode = "FIT2095";
    req.weekNumber = 4;
    next();
});

app.use(express.urlencoded({
    extended: true
}));

app.use(express.json());
```

```
app.get("/", function (req, res) {
    res.sendFile(__dirname + "/index.html");
});
```

```
app.post("/findmax", function (req, res) {
  console.log(req.value1);
  db.push(req.value2);
  db.push(req.value2);

  res.render("response",{
    values:db,
  })
});

app.listen(8080);
```

```
Question 4

Not answered

Not graded
```

Question 5		
Not answered		
Not graded		

Question <b>6</b>	
Not answered	
Not graded	

Not answered Not graded

Question 7

Question 8 Not answered

Not graded

Question 8

■ Week 3: Workshop Quiz

Jump to...

Week 5: Workshop Quiz ▶



```
<u>Dashboard</u> / My units / <u>FIT2095_S2_2021</u> / <u>Week 5</u> / <u>Week 5</u>: <u>Workshop Quiz</u>
```

# Week 5: Workshop Quiz

# Q1)

- 1. MongoDB is a document database. What are 'document' fields, and 'collection' in MongoDB?
- 2. MongoDB document is not a JSON object. Elaborate on this statement.

### **Q2)**

Assume you have the following:

- db is a reference to a database
- 'flights' is a collection with the following fields:
  - o flightDate,
  - o from,
  - o to,
  - o airline,
  - o cost

#### Do the following:

- 1. Retrieve the first 10 flights with 'VA' airline
- 2. Retrieve all flights between 'SA' and 'SYD' in an array format sorted by cost in descending order
- 3. Triple the cost of flights between 'SYD' and 'NT'
- 4. Delete all the flights with cost less than 300

# Q3

Suppose you have the following HTML code:

Using Node.js, Express, and MongoDB, write a server-side code that accepts/handles the form's request and inserts a document into a collection named 'booking' in a database named 'agency' which is hosted by a machine with IP='10.152.168.99' and listens to port 68153.

Last modified: Monday, 23 August 2021, 8:19 AM



<u>Dashboard</u> / My uni	ts / FIT2095_S2_2021 / Assessments / Week 5: Workshop Quiz
Started on	Monday, 23 August 2021, 4:39 PM
State	Finished
Completed on	Monday, 23 August 2021, 5:58 PM
Time taken	1 hour 19 mins
Grade	<b>6.86</b> out of 10.00 ( <b>69</b> %)

### Print friendly format

Question 1
Complete
Mark 6.86 out of 10.00

#### Question 1

MongoDB is a document database. What are 'document' fields, and 'collection' in MongoDB?

A document is a data structure composed of field and value pairs.

A collection is similar to a table in a relational database. It is a set of documents, and you access each document via the collection.

MongoDB document is not a JSON object. Elaborate on this statement.

MongoDB documents are similar to JSON objects.

• Compared to a JSON object, a MongoDB document has support not only for the primitive data types boolean, numbers, and strings, but also other common data types such as dates, timestamps, regular expressions, and binary data.

Comment:

Q1=75% Q2=67% Q3=67%

Question 2
Complete

Not graded

Question 2

Q2. 1.

let query = { airline: /WA/ };

```
db.collection("flights").find(query).limit(10)(function(err, result){
        if(err) throw err;
         console.log(result);
       });
 Question 3
Complete
Not graded
        Question 3
         Q2.2.
        let query = {from: /WA/ or /SYD/ };
        let sortBy = { cost: -1 }
         db.collection ("flights").find (query).sort (sortby).to Array (function (err, result) (sortby)).to Array (function (err, result) (err, result) (sortby)).to Array (function (err, result) (err, result) (sortby)).to Array (function (err, result) (err, result) (err, 
         if(err) throw err;
         console.log(result);
        });
Question 4
Complete
Not graded
         Question 4
         Q2.3
         db.collection("flights").updateOne({ from:'SYD' or 'NT'}, { $mul3: {cost}}, { upsert: true }, function(err, result){
       });
```

```
Question 5
Complete
Not graded
```

Q2.4

```
db.collection("flights").deleteMany({cost: { $lte:300}}, function(err, obj){
  console.log(obj.result);
});
```

```
Question 6
Complete
Not graded
```

```
Question 6
```

```
const mongodb = require('mongodb');
const express = require("express");
const app = express();
app.listen(68153);
const MongoClient = mongodb.MongoClient;
const url = 'mongodb://localhost:68153/';
let db;
MongoClient.connect(url, {useNewUrlParser: true}, function (err, client) {
  if (err) {
    console.log('Err ', err);
    console.log("Connected successfully to server");
    db = client.db('agency');
  }
});
app.post("/postbooking", function (req, res) {
  let bookingDetails = req.body;
  db.collection("users").insertOne({
    source: bookingDetails.bsource,
    target: bookingDetails.btarget,
    airline: bookingDetails.bairline,
  });
  res.redirect("/getbookings");
  app.get("/getbookings", function (req, res) {
    db.collection("agency")
      .find({})
```



```
<u>Dashboard</u> / My units / <u>FIT2095_S2_2021</u> / <u>Week 6</u> / <u>Week 6</u>: <u>Workshop quiz</u>
```

# Week 6: Workshop quiz

Q1)

What are the roles of param in the following statement:

```
module.exports = mongoose.model(param, mySchema);
```

Q2)

Briefly explain how the 'populate()' method works in Mongoose. Support your answer with an example that shows how to invoke it.

Q3)

Assume you have the following schema:

```
const mongoose = require("mongoose");
let doctorSchema = mongoose.Schema({
    _id: {
      type: mongoose.Schema.Types.ObjectId,
      auto: true
    },
    name: {
      firstName: {
        type: String,
        required: true,
      },
      lastName: String,
    }
});
```

Add a validator to the schema such that it only accepts documents that have at least 5 characters in <u>firstName</u>. The message for invalid data is 'firstName length cannot be less than 5'

Q4)

Assume the following:

- you have a database named 'Agency' hosted on a machine with IP=59.0.152.66 with port=88566.
- You have implemented a schema in a file named 'flightschema.js' that represents a collection with five fields: \_id, from, to, airline, and cost

Write a server-side app that uses Mongoose to connect the server to the database and inserts this document: { from: 'MEL', to: 'JNB', airline: 'VA', cost: 2500 }

Last modified: Sunday, 29 August 2021, 11:37 PM



<u>Dashboard</u> / My units / <u>FIT2095_S2_2021</u> / <u>Assessments</u> / <u>Week 6: Workshop Quiz</u>				
Started on	Tuesday, 31 August 2021, 12:53 AM			
State	Finished			
Completed on	Tuesday, 31 August 2021, 1:41 AM			
Time taken	47 mins 52 secs			
Grade	<b>5.38</b> out of 10.00 ( <b>54</b> %)			
Print friendly format				
Question 1				
Complete				
Mark 5.38 out of 10.00				

#### Question 1

Q1.Mongoose will create the database collection for the above model param, params represent the name of the collection for the model

Comment:

Q1=50% Q2=63% Q3=50% Q4=53%

Question **2** Complete

Not graded

Question 2

Q2.

it allows us to reference documents in other collections.

Population is the process of automatically replacing the specified paths in the document with document(s) from other collection(s).

for example, we may populate a single document, multiple documents, a plain object, multiple plain objects, or all objects returned from a query.

```
Question 3
Complete
Not graded
```

#### Question 3

#### Q3.

```
const mongoose = require("mongoose");
let doctorSchema = mongoose.Schema({
   type: mongoose.Schema.Types.ObjectId,
    auto: true
 },
 name: {
    firstName: {
      type: String,
      validate:{
}validator:function(fnLen){
         return minLength.fnLen >=5;
      required: true,
    },
   lastName: String,
 }
});
```

```
Question 4
Complete
Not graded
```

```
Question 4

Q4.

const mongoose = require('mongoose');

const DB = 'mongodb://59.0.152.66:88566/Agency';
```

```
mongoose .connect(DB, {
```

useNewUrlParser: true,

```
})
  .then((con) => {
     console.log('DB connected successfully');
  })
  .catch((err) => {
     console.log(err);
  });
const flightSchema = new mongoose.Schema({
  from: String,
  to: String,
  airline: String,
  cost: Number,
});
const Flight = mongoose.model('Flight', flightSchema);
Flight.create(
  {
    from: 'MEL',
     to: 'JNB',
     airline: 'VA',
     cost: 2500,
```

```
12/11/2021, 22:07
},

(err, data) => {

if (!err) console.log('data saved successfully', data);

else console.log(err);

process.exit(1);
}
```

Question **5**Not answered

);

Not graded

Question 5

Question **6**Not answered
Not graded

Question 7	
Not answered	
Not graded	
Ougation 7	

C	Question 8
١	Not answered
١	Not graded

Question 8

### ■ Week 5: Workshop Quiz

Jump to...

Week 7: Workshop Quiz ▶



```
<u>Dashboard</u> / My units / <u>FIT2095_S2_2021</u> / <u>Week 4</u> /
```

### Week4: Workshop Quiz

# Q1)

Given the following excerpt from a backend server, write a piece of code that is responsible for adding two key-value pairs for every new incoming request. The two pairs are:

unitCode:'FIT2095'

WeekNumber=4

PS: the backtick (``) characters are used to build a string template.

# Q2)

In the following piece of code, briefly explain the role of 'action' and 'method' attributes in the form tag.

```
<html>
<body>
<form action="/data" method="POST">

        User Name:
        <input type="text" name="username" /> </br>
        User Age:
        <input type="number" name="userage" /> </br>
        <button>Submit</button>
</form>
</body>
</html>
```

# Q3)

Rewrite app.js such that

- 1. Extracts value1 and value2 from the POST request that is sent by index.html on pathname='/findmax'
- 2. Responds back after rendering 'response.html'. The template shows the maximum value between value1 and value2

```
const express = require("express");
const app = express();
// your code goes here
app.use(function (req, res, next) {
    req.unitCode = "FIT2095";
   req.weekNumber = 4;
   next();
});
app.use(express.urlencoded({
    extended: true
}));
app.use(express.json());
//your code here
app.get("/", function (req, res) {
    res.sendFile(__dirname + "/index.html");
// your code here
app.listen(8080);
```

index.html

response.html

Last modified: Monday, 16 August 2021, 9:06 PM

■ VS Code keyboard-shortcuts-macos

```
Jump to...
```



Dashboard / My units / FIT2095 S2 2021 / Assessments / Week 4: Workshop Quiz

Started on Tuesday, 17 August 2021, 1:47 AM

State Finished

Completed on Tuesday, 17 August 2021, 2:30 AM

Time taken 43 mins 1 sec

Grade 7.20 out of 10.00 (72%)

### Print friendly format

Question 1

```
Question 1
Complete
Mark 7.20 out of 10.00
```

```
Question 2
Complete
Not graded
```

#### Question 2

Q1=100% Q2=100% Q3=36%

action: another name for pathname method: tells that a parse request is needed to be sent

```
Question 3
Complete
Not graded
```

```
const express = require("express");
const app = express();

let db=[];

app.use(function (req, res, next) {
    req.unitCode = "FIT2095";
    req.weekNumber = 4;
    next();
});

app.use(express.urlencoded({
    extended: true
}));

app.use(express.json());
```

```
app.get("/", function (req, res) {
    res.sendFile(__dirname + "/index.html");
});
```

```
app.post("/findmax", function (req, res) {
  console.log(req.value1);
  db.push(req.value2);
  db.push(req.value2);

  res.render("response",{
    values:db,
  })
});

app.listen(8080);
```

```
Question 4

Not answered

Not graded
```

Question 5		
Not answered		
Not graded		

Question <b>6</b>	
Not answered	
Not graded	

Not answered Not graded

Question 7

Question 8 Not answered

Not graded

Question 8

■ Week 3: Workshop Quiz

Jump to...

Week 5: Workshop Quiz ▶





# Question 1 (15 marks)

Assume **Student** is a reference to a Mongoose model that has the following field:

```
teachers: [{
    type: mongoose.Schema.Types.ObjectId,
    ref: 'Teacher'
```

Assume you have the following API endpoint:

```
app.put('/students/:sId/teachers/:tId',addTeacher);
```

Where sld is the student's ID, tld is the teacher's ID, and addTeacher is the callback function)

Develop the function addTeacher that pushes the teacher's id tld into the array 'teachers' for a student with ID equals to sld. The function must respond in JSON format with the error of an error occurs, "Done" otherwise.



# Question 2 (12 marks)

The following piece of code is designed to add a new document of type Teacher to a collection.

It is supposed to accept the following request, but the code didn't work:

http://localhost:8080/teachers?fullName=Time&school=Primary1

Find all the possible problems in the code above and suggest solutions.

Notes: Assume all the variables (app, Teacher, mongoose) are well implemented and/or referenced.



# Question 3 (5+2=7 marks)

- a) In which scenario would you use PUT http request instead of POST?
- b) RESTFul API is a stateless architecture. Elaborate this statement.



```
Dashboard / My units / FIT2095_S2_2021 / Week 8 / Workshop Quiz: Week 8:
```

## Workshop Quiz: Week 8:

## Question 1 (10 marks)

Assume you have an Angular app with a single component.

The source code of app.component.ts is listed below:

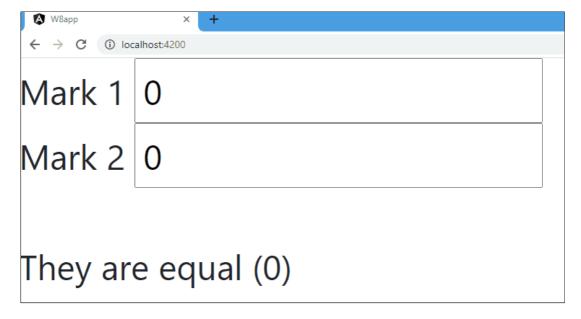
```
import { Component } from '@angular/core';

@Component({
    selector: 'app-root',
    templateUrl: './app.component.html',
    styleUrls: ['./app.component.css']
})

export class AppComponent {
    no1 = 0;
    no2 = 0;
}
```

Code app.component.html such that it takes two numbers and displays which one is the maximum or if they are equal as shown below.

Note: you must NOT make any changes to app.component.ts file.



Assume you are developing an Angular App. Write the simplest component code that is sufficient to produce the following output. You need to provide the source codes of app.component.html and app.component.ts.

The component must take as input:

- a string represents the fruit's name
- · a number represents the fruit's calories

Two buttons should be added:

- · save a new fruit
- delete unhealthy fruits (any fruit with more than 100 calories)



Question 3 (5 marks)

Briefly explain why is TypeScript considered a strongly typed JavaScript? Support your answer with an example.

Last modified: Monday, 13 September 2021, 12:46 PM



Started on Tuesday, 14 September 2021, 5:11 AM

State Finished

Completed on Tuesday, 14 September 2021, 5:13 AM

Time taken 1 min 49 secs

Grade 7.43 out of 10.00 (74%)

#### Print friendly format

Question 1
Complete
Mark 7.43 out of 10.00

#### Question 1

Mark 1:
<input [(ngModel)]="no1" />
<br>
<br>
<br>
<br>
Mark 2:
<input [(ngModel)]="no2" />
<br>
<br>
<br>
<div \*ngIf="no1=no2">They are equal {{no1}}</div>
<div \*ngIf="no1>no2"> Mark 1 is maximum {{no1}}</div>
<div \*ngIf="no2>no1"> Mark 2 is maximum {{no1}}</div>
<div \*ngIf="no2>no1"> Mark 2 is maximum {{no1}}</div>
<div \*ngIf="no2>no1"> Mark 2 is maximum {{no1}}</div>
</div

Comment:

Q1=80% Q2=75% Q3=60%

Question **2** Complete

Not graded

#### Question 2

Q2.

#### app.components.ts

```
import { Component } from '@angular/core';
@Component({
 selector: 'app-root',
 templateUrl: './app.component.html',
 styleUrls: ['./app.component.css']
export class AppComponent {
 data :any[]= [];
 Fruit: string = ";
 Calories: number = 0;
 addFruit() {
  let newFruitObj={
   Fruit:this.Fruit,
   Calories:this.Calories,
  }
  this.data.push(newFruitObj);
 deleteFruit() {
  this.data = this.data.filter(Fruit => (Fruit["calories"] < 100));
}
```

Question **3**Complete

Not graded

#### Question 3

#### Q2. app. component.html

```
<div>
Enter Your Fruit:
<input [(ngModel)]="Fruit" />
<br>
<input [(ngModel)]="Calories" />
<button (click)="addFruit()">Add Fruit</button>
<button (click)="deleteFruit()">Delete Fruit</button>
<br>
<body>
{{item.Fruit}}
{{item.Calories}}
</body>
</div>
```

Question **4**Complete

Not graded

Q3.

Browser doesn't understand typescript. Typescript is able to transfer the code into javascript code and send it back to the browser which is safe. And it is able to use different features. For example, variables and other data structures can be declared to be of a specific type, like a string or a boolean, and typescript is going to check their values.

Question 5

Not answered

Not graded

Question 5

Question 6

Not answered

Not graded

Question <b>7</b> Not answered Not graded			
Question 7			

Question 8		
Not answered		
Not graded		

■ Week 7: Workshop Quiz

Jump to...

Week 9: Workshop Quiz ▶



```
Dashboard / My units / FIT2095_S2_2021 / Week 9 / Week 9 workshop quiz
```

## Week 9 workshop quiz

- Q1) What is observable in Angular? Support your answer with an example. (5m)
- Q2) What is the role of declarations, imports, providers and bootstrap in Angular module? (8m)
- Q3) What is the role of the attribute selector in Angular component? (2m)
- Q4) Assume you have the following function that is implemented in a service injected to a component called 'Weather'. (7m)

```
getTemperature() {
   return this.http.get('/temp');
}
```

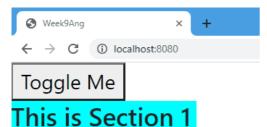
The component has a variable called temperature and it is declared as follows:

```
temperature: number;
```

Develop a function called getCurrentTemperature for the component that should invoke getTemperature() and assign the returned data to temperature.

Q5) Assume you have a component called 'Student' that has two div sections and a button in its template. (12m)

What should you do to make the button "Toggle Me" works like a toggle button such that it displays one div (section) at a time as shown below. Your answer should provide all the required source codes to achieve this task.





<u>Dashboard</u> / Miy um	ts / FIT2095_S2_2021 / Assessments / Week 9: Workshop Quiz
Started on	Monday, 20 September 2021, 8:05 PM
State	Finished
Completed on	Monday, 20 September 2021, 11:32 PM
Time taken	3 hours 27 mins
Grade	<b>6.76</b> out of 10.00 ( <b>68</b> %)
rint friendly format	
Question <b>1</b>	
Complete	
Mark 6.76 out of 10.00	

Question 1

q1.

Angular makes use of observables as an interface to handle a variety of common asynchronous operations. For example:

The Router and Forms modules use observables to listen for and respond to user-input events.

Comment:

Q1=60% Q2=100% Q3=100% Q4=71% Q5=42%

Question **2**Complete

Not graded

Question 2

q2.

#### **Declarations:**

are to make directives (including components and pipes) from the current module available to other directives in the current module.

٠						
ı	m	n	n	-	te	. '

makes the exported declarations of other modules available in the current module

#### providers:

are to make services and values known to DI (dependency injection).

#### bootstrap:

a function component in the core ng module that is used for starting up the Angular application manually, which gives you more control over how you initialize your application.

Question **3**Complete
Not graded

#### Ouestion 3

q3.

A selector is one of the properties of the object that we use along with the component configuration and it is used to identify each component uniquely, it also defines how the current component is represented in the HTML DOM.

Question 4
Complete
Not graded

#### Question 4

q4.

getCurrentTemperature(){
this.dbService.getTemperature().subscribe((data:number)=>{
this.temperature=data;
});
}

```
Question 5
Complete
Not graded
```

q5.

- create a function that recognize the id we put in div section and change the display
- · call the function in the button's code

```
<body>
<button onclick="changeDisplay()">Toogle Me</button>
 <div id="cd1" style="background-color: aqua;width:20%">
 <h5>This is Section 1</h5>
 </div>
 <div id="cd2" style="background-color: chocolate;width:20%">
 <h5>This is Section 2</h5>
 </div>
var b=document.getElementById("cd2");
 y.style.display="none";
function changeDisplay(){
var x=document.getElementById("cd1");
 if(a.style.display=="none")
 a.style.display="block";
 b.style.display="none";
 }
 else
 {
 a.style.display="none";
 b.style.display="block";
```

```
Question 6
Not answered
Not graded
```

#### Question 6

}

Question <b>7</b> Not answered Not graded		
Question 7		

Not answered

Not graded

Question 8

■ Week 8: Workshop Quiz

Jump to...

Week 10: Workshop Quiz ▶



Dashboard / My units / FIT2095\_S2\_2021 / Week 10 / Week 10 workshop quiz

## Week 10 workshop quiz

- Q1) Suppose you are working on an Angular application, briefly explain the difference between 'routerLink' and 'href' attributes in an 'a' element. (4m)
- Q2) Briefly explain one difference and one similarity between 'Pipe' and 'Service' in Angular. (6m)
- Q3) What should you do to navigate from component A to component B using typescript language? (6m)
- Q4) Suppose you are working on an Angular application that contains the following components: (15m)
- SignupComponent
- · DashboardComponent
- InvalidComponent

Develop the array of Routes that will be fed as input to RouterModule.forRoot() function in 'app.modules.ts'. Notes:

- · Design your own paths
- · Redirect empty paths to DashboardComponent
- · Redirect unknown paths to InvalidComponent
- Q5) Explain why and how PWA applications are more responsive than non-PWA applications and can work with offline connections. (6m)

Last modified: Monday, 4 October 2021, 10:36 PM



<u>Dashboard</u> / My uni	ts / FIT2095_S2_2021 / Assessments / Week 10: Workshop Quiz
Started on	Tuesday, 5 October 2021, 2:59 AM
State	Finished
Completed on	Tuesday, 5 October 2021, 4:45 AM
Time taken	1 hour 45 mins
Grade	<b>8.38</b> out of 10.00 ( <b>84</b> %)

#### Print friendly format

Question 1
Complete

Mark 8.38 out of 10.00

#### Question 1

Q1)

Herf attribute is a basic attribute provided by HTML for navigation between pages and reloading the page on click. routerLink is provided by angular to navigate between different components without reloading.

- herf ends the state of the original page
- routerLink keeps the state of the original page

Comment:

Q1=100% Q2=100% Q3=67% Q4=80% Q5=83%

Question **2** Complete

Not graded

#### Question 2

Q2)

• one difference: Angular pipes are functions that are used to transform data in **Angular templates** and an angular service is an object that organize and share methods,models or data **between different components**.

• one similarity: pipe and service are both required a one-time declaration/instantiation and they are both useable throughout the application

```
Question 3
Complete
Not graded
 Question 3
 Q3)
 You should enable angular routing to navigate from component A to component B
     1. import the RouterModule and provide the appRoutes array as input
     2. import the Routes and RouterModule from "@angular/router":
 import{ ComponentA} from "./A.component";
 import{ ComponentB } from "./B.component";
 const appRoutes: Routes =[
 { path: "componenta", component: ComponentA },
 { path:"", redirectTo:"/componentb", pathMatch:"full"},
 ];
Question 4
Complete
Not graded
```

#### Ouestion 4

```
Q4)
```

```
import{ AppComponent } from "./app.component";
import{ DatabaseService } from "./database.service";
import{ HttpClientModule } from "@angular/common/http";
import{ FormsModule } from "@angular/forms";
import{ SignupComponent } from "./signup/signup.component";
import{ DashboardComponent } from "./dashboard/dashboard.component";
import{ InvalidComponent } from "./invalid/invalid.component";
import{ RouterModule, Routes } from "@angular/router";

const appRoutes: Routes =[
{ path: "signup", component: SignupComponent },
```

Question 5  Q5)  PWAapplications are faster than non-PWA applications, it also reduces data needs with fewer API requests, and it leads to better use experience and better engagement.  PWA allows users to interact with the app independent of the connection. To make it work offline, you need to store the static files necessary to run the app. PWA catches the data ahead of time and it is done by the use of what is known as a service worker. The service worker offers a programmatic method of cache resources.	t patri. "", component: in	wolidComponent) //wildoord route for a invalid ness
Question 5  Question 5  Q5)  PWAapplications are faster than non-PWA applications, it also reduces data needs with fewer API requests, and it leads to better use experience and better engagement.  PWA allows users to interact with the app independent of the connection. To make it work offline, you need to store the static files necessary to run the app. PWA catches the data ahead of time and it is done by the use of what is known as a service worker. The service worker offers a programmatic method of cache resources.	1.	ivalidComponent}, // wildcard route for a invalid page
Ouestion 5  Q5)  PWAapplications are faster than non-PWA applications, it also reduces data needs with fewer API requests, and it leads to better use experience and better engagement.  PWA allows users to interact with the app independent of the connection. To make it work offline, you need to store the static files necessary to run the app. PWA catches the data ahead of time and it is done by the use of what is known as a service worker. The service worker offers a programmatic method of cache resources.	1,	
Ouestion 5  Q5)  PWAapplications are faster than non-PWA applications, it also reduces data needs with fewer API requests, and it leads to better use experience and better engagement.  PWA allows users to interact with the app independent of the connection. To make it work offline, you need to store the static files necessary to run the app. PWA catches the data ahead of time and it is done by the use of what is known as a service worker. The service worker offers a programmatic method of cache resources.		
Ouestion 5  Q5)  PWAapplications are faster than non-PWA applications, it also reduces data needs with fewer API requests, and it leads to better use experience and better engagement.  PWA allows users to interact with the app independent of the connection. To make it work offline, you need to store the static files necessary to run the app. PWA catches the data ahead of time and it is done by the use of what is known as a service worker. The service worker offers a programmatic method of cache resources.	Nucetion 5	
Question 5  Q5)  PWAapplications are faster than non-PWA applications, it also reduces data needs with fewer API requests, and it leads to better use experience and better engagement.  PWA allows users to interact with the app independent of the connection. To make it work offline, you need to store the static files necessary to run the app. PWA catches the data ahead of time and it is done by the use of what is known as a service worker. The service worker offers a programmatic method of cache resources.		
PWAapplications are faster than non-PWA applications, it also reduces data needs with fewer API requests, and it leads to better use experience and better engagement.  PWA allows users to interact with the app independent of the connection. To make it work offline, you need to store the static files necessary to run the app. PWA catches the data ahead of time and it is done by the use of what is known as a service worker. The service worker offers a programmatic method of cache resources.	Not graded	
PWAapplications are faster than non-PWA applications, it also reduces data needs with fewer API requests, and it leads to better use experience and better engagement.  PWA allows users to interact with the app independent of the connection. To make it work offline, you need to store the static files necessary to run the app. PWA catches the data ahead of time and it is done by the use of what is known as a service worker. The service worker offers a programmatic method of cache resources.		
<ul> <li>PWAapplications are faster than non-PWA applications, it also reduces data needs with fewer API requests, and it leads to better use experience and better engagement.</li> <li>PWA allows users to interact with the app independent of the connection. To make it work offline, you need to store the static files necessary to run the app. PWA catches the data ahead of time and it is done by the use of what is known as a service worker. The service worker offers a programmatic method of cache resources.</li> </ul>	Question 5	
<ul> <li>PWAapplications are faster than non-PWA applications, it also reduces data needs with fewer API requests, and it leads to better use experience and better engagement.</li> <li>PWA allows users to interact with the app independent of the connection. To make it work offline, you need to store the static files necessary to run the app. PWA catches the data ahead of time and it is done by the use of what is known as a service worker. The service worker offers a programmatic method of cache resources.</li> </ul>	OE)	
experience and better engagement.  PWA allows users to interact with the app independent of the connection. To make it work offline, you need to store the static files necessary to run the app. PWA catches the data ahead of time and it is done by the use of what is known as a service worker. The service worker offers a programmatic method of cache resources.	(3)	
experience and better engagement.  PWA allows users to interact with the app independent of the connection. To make it work offline, you need to store the static files necessary to run the app. PWA catches the data ahead of time and it is done by the use of what is known as a service worker. The service worker offers a programmatic method of cache resources.	PWAapplications are f	faster than non-PWA applications, it also reduces data needs with fewer API requests, and it leads to better uso
necessary to run the app. PWA catches the data ahead of time and it is done by the use of what is known as a service worker. The service worker offers a programmatic method of cache resources.  Question 6  Not answered  Not graded	experience and better	r engagement.
service worker offers a programmatic method of cache resources.  Question 6 Not answered Not graded		
Question <b>6</b> Not answered Not graded		
Not answered Not graded		
Not answered Not graded		
Not answered Not graded		
Not answered Not graded		
Not answered Not graded		
Not answered Not graded		
Not answered Not graded		
Not answered Not graded		
	Question <b>6</b>	
Question 6		
Question 6	Question <b>6</b> Not answered Not graded	
	Not answered	
	Not answered	
	Not answered	
	lot answered lot graded	
	lot answered lot graded	
	ot answered ot graded	
	lot answered lot graded	
	ot answered ot graded	
	lot answered lot graded	
	Not answered	
	Not answered	
	Not answered Not graded	
	Not answered	
	Not answered Not graded	

Not answered Not graded

Question 8	
Not answered	
Not graded	

### ■ Week 9: Workshop Quiz

Jump to...

Week 11: Workshop Quiz ▶



Dashboard / My units / FIT2095\_S2\_2021 / Week 11 / Week 11- workshop quiz

## Week 11- workshop quiz

- Q1) What are the datatype and the role of param1 and param2? Provide an example. (6m) io.sockets.emit(param1,param2);
- Q2) What are the datatype and the role of param3 and param4? Provide an example. (6m) io.on(param3,param4);
- Q3) Suppose you are developing a Node.js application that uses Google Text-to-Speech service. Briefly explain how does your application use the key JSON file to access Google cloud services. (5m)
- Q4) Develop a server-side application using Node.js and Socket.io that: (14m)
  - 1. Listens to all socket.io connection requests on port 9595
  - 2. With each new connection, the server broadcasts to all connected clients except the sender the id of the new socket using an event named 'newSocketID'.
  - 3. Listens to a socket event named 'values', which arrives with an object contains two integers no1 and no2.
  - 4. Calculates result=no1-no2
  - 5. Broadcasts a new object contains the three values no1, no2 and result to all clients including the sender using an event named 'newResult'.

Last modified: Monday, 11 October 2021, 4:50 PM



<u>Dashboard</u> / My uni	ts / FIT2095_S2_2021 / Assessments / Week 11: Workshop Quiz
Started on	Monday, 11 October 2021, 9:26 PM
State	Finished
Completed on	Monday, 11 October 2021, 10:33 PM
Time taken	1 hour 7 mins
Grade	<b>7.74</b> out of 10.00 ( <b>77</b> %)

#### Print friendly format

Question 1	
Complete	
Mark 7.74 out of 10.00	

#### Question 1

Q1)

param1- datatype string, role: name of an event param2-datatype role: data to be sent

example: io.sockets.emit('user connect',user+'connect to the chatroom');

Comment:

Q1=67% Q2=100% Q3=100% Q4=64%

Question 2
Complete
Not graded

Question 2

q2)

param3- datatype - string, role: an event that represents the connection param4- datatype -function role: receive sockets parameters

```
example: io.on(connection', function(soccket)=>{
console.log(socket);
})
```

Question 3 Complete Not graded

#### Ouestion 3

Q3)

- 1.Set up authentication by going to create a service account key in the GCP console
- 2.Set the environment variable GOOGLE\_APPLICATION\_CREDENTIALS to the file path of the JSON file that contains your service account key.
- 3. Enable the Cloud Text-to-Speech API
- 4.Install and initialize the Cloud SDK.

```
Question 4
Complete
Not graded
```

#### Question 4

```
Q4.
```

```
let express = require("express");
let path = require("path");
let app = express();
let server = require("http").Server(app);
let io = require("socket.io")(server);
let port =9595;
app.use("/", express.static(path.join(\_dirname, "dist/chatApp")));\\
io.on("connection", socket =>{
console.log("new connection made from client with ID="+socket.id);
```

#### //to all the clients except the sender

socket. broadcast. emit("newSocketID", socket.id);

```
socket.on("values", function(data){
console.log("got data from values ");
}
//to all connected clients
io.emit('newResult');
server.listen(port,()=>{
console.log("Listening on port "+ port);
});
```

Not answered

Not graded

Question 5

Question 6

Not answered

Not graded

Question 7		
Not answered		
Not graded		

Question 8		
Not answered		
Not graded		

Question 8

#### ■ Week 10: Workshop Quiz

Jump to...

Week 12: Workshop Quiz ▶



Dashboard / My units / FIT2095\_S2\_2021 / Week 12 / Week 12 Workshop Quiz Questions

## Week 12 Workshop Quiz Questions

- Q1) Assume you have a collection that contains documents with attributes 'id' and 'age'. Give a scenario of NoSQL Injection attack that retrieves objects from the database. Suggest mitigation for your scenario. (6m)
- Q2) Briefly describe two advantages and disadvantages of allowing users to log in to a MEAN stack application using their Facebook account. (8m)
- Q3) Suppose you are building a chat application using the node.js, briefly describe two advantages and two disadvantages of using Firebase instead of MongoDB. (8m)

Last modified: Monday, 18 October 2021, 3:27 PM



```
Started on Tuesday, 19 October 2021, 6:06 AM

State Finished

Completed on Tuesday, 19 October 2021, 6:31 AM

Time taken 25 mins 6 secs

Grade 8.64 out of 10.00 (86%)

Print friendly format
```

```
Question 1
Complete
Mark 8.64 out of 10.00
```

#### Question 1

Q1) Assume you have a collection that contains documents with attributes 'id' and 'age'. Give a scenario of NoSQL Injection attack that retrieves objects from the database. Suggest mitigation for your scenario. (6m) db.collection("users").findOne(
{
 userID: String(ID),
 userAge: String(age),
 },
 function(err, result){
 if(result !==null){
 res.redirect("/");
 }else{
 res.redirect("/details");
 }
 }
}

Comment:
Q1=50% Q2=100% Q3=100%

```
Question 2
Complete
Not graded
```

711/2021, 22.10	Week 12. Workshop Quiz. Attempt review
Q2) Briefly describe two advantages using their Facebook account. (8m)	and disadvantages of allowing users to log in to a MEAN stack application
advantages:1.lt could be more convenient	for the users(e.g. registering)
2. less fail login	
disadvantages: 1.there is no control over t	he login process
2.user's information are compromised if th	
Question 3	
Complete	
Not graded	
disadvantages of using Firebase ins advantages:1. it connects with Google clou 2. it offers a faster CDN for static website disadvantages:1. MongoDB is secure, no S	ud storage and users can use google cloud storage API to manipulate the objects
Question 4	
Not answered	
Not graded	

Question 5	
Not answered	
Not graded	
Question 5	
Question <b>6</b>	
Not answered  Not graded	
Not graded	
Question 6	
o	
Question 7	
Not answered	
Not graded	

https://lms.monash.edu/mod/quiz/review.php?attempt=22116851&cmid=8806742

Question 8		
Not answered		
Not graded		

■ Week 11: Workshop Quiz

Jump to...

Week 2 Pre-reading Quiz ▶