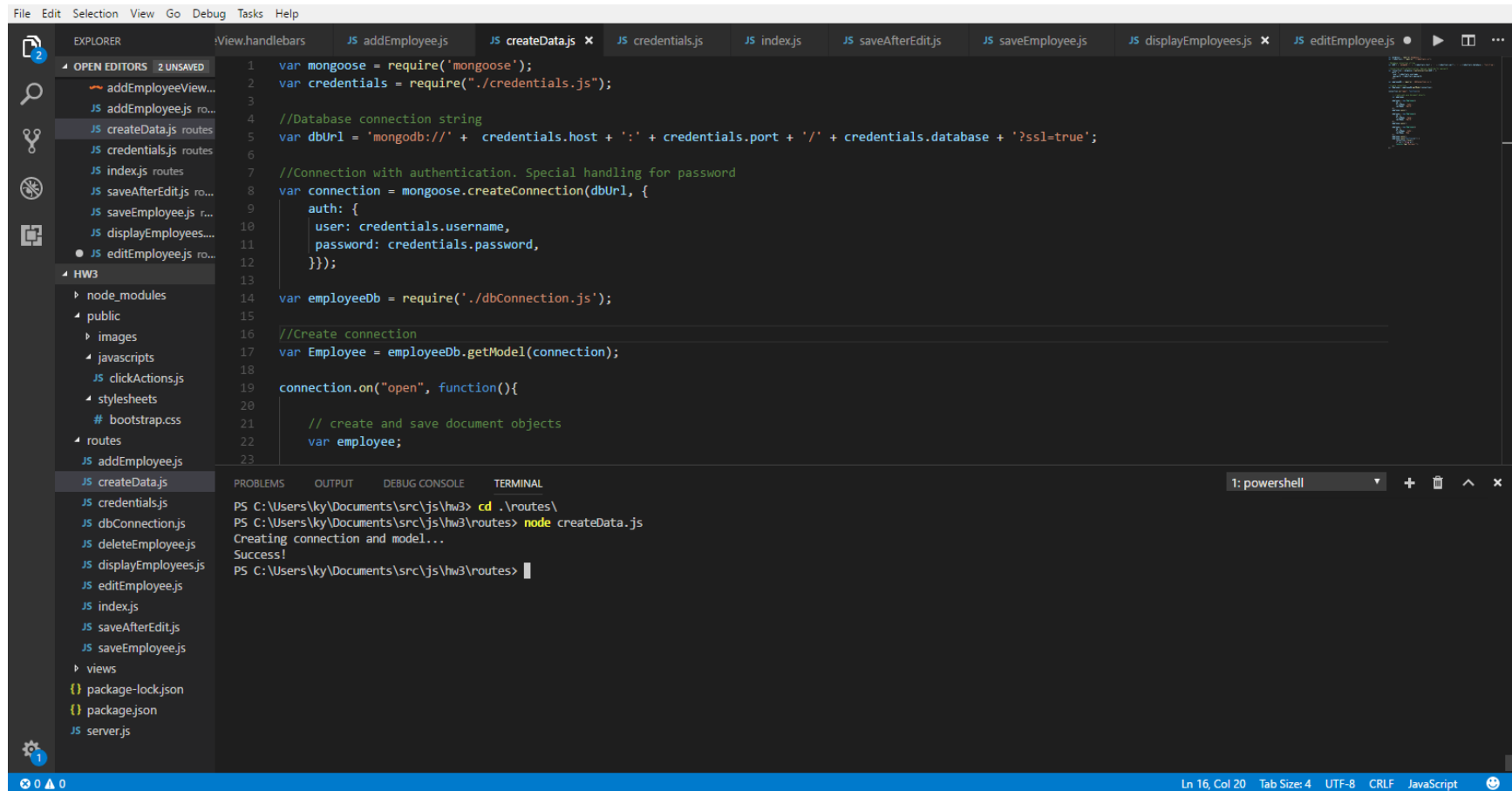


Part 1. Populate Database

./routes/createData.js is designed for create initial documents in mongoDB

*Execute this script after documents already exist in database creates duplicates



The screenshot shows the Visual Studio Code editor with the file explorer on the left. The file explorer shows the project structure with folders like node_modules, public, images, javascripts, stylesheets, bootstrap.css, routes, and views. The routes folder is expanded, showing files like addEmployee.js, createData.js, credentials.js, dbConnection.js, deleteEmployee.js, displayEmployees.js, editEmployee.js, index.js, saveAfterEdit.js, saveEmployee.js, package-lock.json, package.json, and server.js. The createData.js file is selected and its content is displayed in the editor. The code in createData.js is as follows:

```
1 var mongoose = require('mongoose');
2 var credentials = require("../credentials.js");
3
4 //Database connection string
5 var dbUrl = 'mongodb://' + credentials.host + ':' + credentials.port + '/' + credentials.database + '?ssl=true';
6
7 //Connection with authentication. Special handling for password
8 var connection = mongoose.createConnection(dbUrl, {
9   auth: {
10     user: credentials.username,
11     password: credentials.password,
12   }
13 });
14
15 var employeeDb = require('./dbConnection.js');
16
17 //Create connection
18 var Employee = employeeDb.getModel(connection);
19
20 connection.on("open", function(){
21   // create and save document objects
22   var employee;
23
24   // ... (rest of the code is truncated in the image) ...
25 });
```

The terminal window at the bottom shows the command prompt with the following commands and output:

```
PS C:\Users\ky\Documents\src\js\hw3> cd ./routes\
PS C:\Users\ky\Documents\src\js\hw3\routes> node createData.js
Creating connection and model...
Success!
PS C:\Users\ky\Documents\src\js\hw3\routes>
```

The status bar at the bottom indicates the current line is 16, column 20, with a tab size of 4, UTF-8 encoding, CRLF line endings, and the file is a JavaScript file.

After initialization, document appears in database.

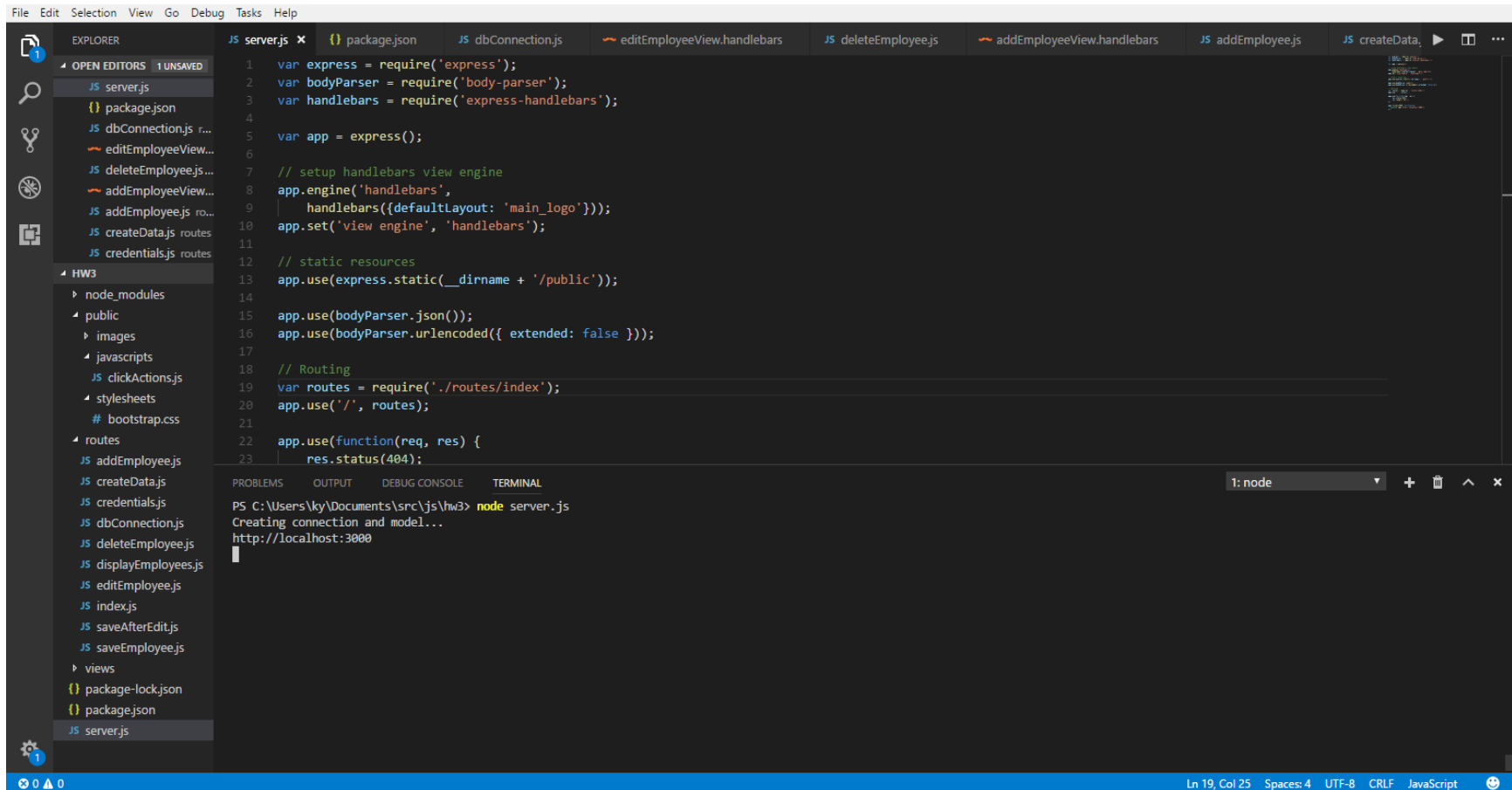
*Database is cloud hosting, should be accessible regardless where the application is executed.

The screenshot displays the Azure Cosmos DB Data Explorer interface for an account named 'cs602hw3'. The left sidebar contains navigation options: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Quick start, and Data Explorer (which is selected). Below these are settings for Connection String, Preview Features, Replicate data globally, Default consistency, Firewall, and Locks. The main pane shows the 'COLLECTIONS' list with 'cs602db' expanded to show 'employeeemodels'. Under 'employeeemodels', 'Documents' is selected, displaying a list of document IDs (e.g., '5a79194b8ba157236c...'). A 'Load more' link is visible below the list. The right pane shows the 'Mongo Docu...' tab with a 'Scale & Settings' sub-tab. It includes buttons for 'New Document', 'Update', 'Discard', and 'Delete'. Below these, it states 'No filter applied' with an 'Edit Filter' button. The document content is displayed in a JSON format:

```
1 {  
2   "_id" : ObjectId("5a79194b8ba157236ceebc69"),  
3   "Id" : 1,  
4   "FirstName" : "John",  
5   "LastName" : "Smith",  
6   "__v" : 0  
7 }
```

Part 2. Web Application

Server can be started by execute server.js



The screenshot shows the Visual Studio Code editor interface. The Explorer sidebar on the left displays a file tree for a project named 'HW3'. The file 'server.js' is selected and open in the main editor. The code in 'server.js' is as follows:

```
1 var express = require('express');
2 var bodyParser = require('body-parser');
3 var handlebars = require('express-handlebars');
4
5 var app = express();
6
7 // setup handlebars view engine
8 app.engine('handlebars',
9   handlebars({defaultLayout: 'main_logo'}));
10 app.set('view engine', 'handlebars');
11
12 // static resources
13 app.use(express.static(__dirname + '/public'));
14
15 app.use(bodyParser.json());
16 app.use(bodyParser.urlencoded({ extended: false }));
17
18 // Routing
19 var routes = require('./routes/index');
20 app.use('/', routes);
21
22 app.use(function(req, res) {
23   res.status(404);
24 });
```

The terminal at the bottom shows the command `node server.js` being executed in a PowerShell window. The output indicates that the connection and model are being created, and the server is running on `http://localhost:3000`.

```
PS C:\Users\ky\Documents\src\js\hw3> node server.js
Creating connection and model...
http://localhost:3000
```

The status bar at the bottom of the editor shows the current line and column as `Ln 19, Col 25`, and the file encoding as `UTF-8`.

Home Page, routed to /employees and displaying all available employees from database

localhost:3000/employees

Apps

PassLeader Premium

Cashback Rebates Co

Jenkins [Jenkins]

BU

Make your own Amii

Breath of the Wild W

BU Student Central | Bos

LFCS Study Guide

LFCS: How to use GN

BOSTON
UNIVERSITY

List of Employees

Id	First Name	Last Name	
1	John	Smith	<a>Edit <a>Delete
2	Jane	Smith	<a>Edit <a>Delete
3	John	Doe	<a>Edit <a>Delete

Add

Adding new employee

localhost:3000/employees/add

BOSTON UNIVERSITY

Add an employee

Id:

First Name:

Last Name:

After Add

localhost:3000/employees

BOSTON UNIVERSITY

List of Employees			
Id	First Name	Last Name	
1	John	Smith	Edit Delete
2	Jane	Smith	Edit Delete
3	John	Doe	Edit Delete
5	Joe	Smith	Edit Delete

Edit Employee

BOSTON
UNIVERSITY

Edit Employee

Id:

5

First Name:

Joe5

Last Name:

Smith

Save

Cancel

After Edit

← → ↺

localhost:3000/employees

Apps

PassLeader Premium

Cashback Rebates Co

Jenkins [Jenkins]

BU

Make your own Amii

Breath of the Wild W

Student Central | Bos

LFCS Study Guide

LFCS: How to use GN

List of Employees			
Id	First Name	Last Name	
1	John	Smith	Edit Delete
2	Jane	Smith	Edit Delete
3	John	Doe	Edit Delete
5	Joe5	Smith	Edit Delete

Add

Delete Employee

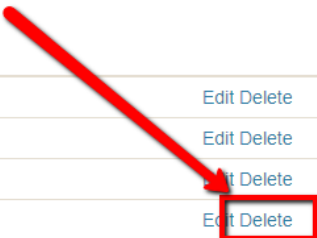
← → ↻ ⓘ localhost:3000/employees

Apps PassLeader Premium Cashback Rebates Co Jenkins [Jenkins] BU Make your own Amitt 20 Breath of the Wild W BU Student Central | Bos LFCS Study Guide LFCS: How to use GN

BOSTON UNIVERSITY

Id	First Name	List of Employees		
		Last Name		
1	John	Smith		Edit Delete
2	Jane	Smith		Edit Delete
3	John	Doe		Edit Delete
5	Joe5	Smith		Edit Delete

Add



After Delete

← → ↻ ⓘ localhost:3000/employees

Apps PassLeader Premium Cashback Rebates Co Jenkins [Jenkins] BU Make your own Amitt 20 Breath of the Wild W BU Student Central | Bos LFCS Study Guide LFCS: How to use GN

BOSTON UNIVERSITY

Id	First Name	List of Employees		
		Last Name		
1	John	Smith		Edit Delete
2	Jane	Smith		Edit Delete
3	John	Doe		Edit Delete

Add