README

HOM/1079/16 - Junior Developer – Assessment

Ref – 1501637

About the application.

The application was built using the Node.js Tools for Visual Studio 2015 IDE due to my familiarity with the software from previous experience and the toolsets it provides.

The application uses Node.js and Express to provide dynamic pages driven by templates to display job applicants and their details. Information for the jobs and applicants is stored using MongoDB chosen for its scalability and high performance when handling large data sets. JavaScript and JQuery has been used for frontend validation on forms and Jasmine has been used alongside Request for testing of functions in the Node.Js environment.

The project files have been included.

Installation

Node.js

Node.js can be downloaded from

https://nodejs.org/en/download/current/

Download and install the latest executable.

MongoDB

MongoDB can be downloaded from

http://www.mongodb.org/downloads

Download and install the latest executable, create the following folders on the root of your C:\ drive.

```
\data
```

```
\data\db
```

Navigate to the installation directory of MongoDB, the default Windows installation path is

```
C:\Program Files\MongoDB\Server\<Version Number>\bin
```

From there launch a command prompt and execute

```
mongod.exe
```

This is the server executable, the database will now be running, the client for the database can be launched by opening another command prompt from the same folder and executing

```
mongo.exe
```

This is the client for the database, commands can be issued from this window to the database. Create a database by typing the command

```
Use <Database Name>
```

Install Project from GitHub

Install the application directly from GitHub, do this using the Node.js executable and installing via Node Package Manager using the command

```
npm install LucasCairns/HOMApplication –g
```

The package will download and install on your machine, you can find the installation directory of the application by using the command.

```
npm root –g
```

Navigate to this directory using the command

```
cd <Address>
```

When at the directory use the following command to initialise the server

```
npm start
```

Once the server has started navigate to the following to view the application.

```
http://localhost:8080
```

Configuration

The application can be configured by navigating to the root directory and editing the config.json file.

```
{
  "port": 8080,
  "databaseUrl": "mongodb://localhost:27017/HOMApplication",
  "databaseJobCollection": "jobs",
  "databaseApplicantCollection": "applicants",
  "testEnvironment": false
}
```

The properties include, port number, database server URL, database collections and a flag that can be set to inject test data into the database and due to how MongoDB works create the collections at the same time. Note when changing the database variables that MondoDB is case sensitive. The ‘testEnvironment’ flag reverts back to false after the data has been injected, the data can be removed from the database using the following commands in the MongoDB client to remove the collections.

```
db.use <DatabaseName>

db.<Collection Name>.drop()
```

Database structure

Below are tables displaying the structure of the database collections, using MongoDB allowed for great flexibility with the way data was stored in the database, allowing for the employment history of each applicant to be contained in an array of objects. This, combined with the way in which MongoDB returned information proved useful especially when developing the Applicant Details page.

Jobs Collection

ID	**Title**	**CloseDate**	**Applicants**
AAA/1234/12	Text	Date	[Array of Applicant ID]

Applicant Collection

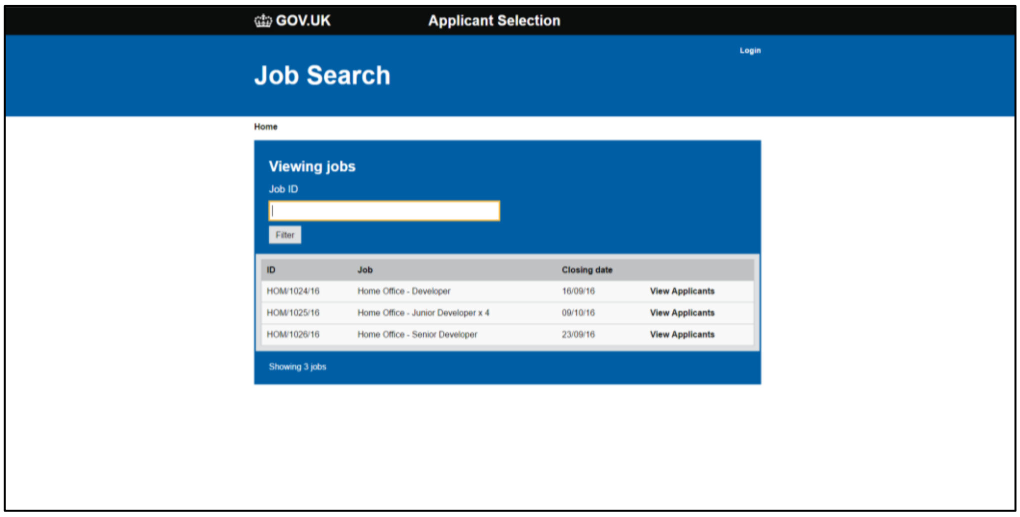
ID	**Name**	**Surname**	**DOB**	**Employment**
AAA/123456	Text	Text	Date	[Array of JSON]

Employment

Employer	**StartDate**	**EndDate**	**Role**	**Description**
Text	Date	Date	Text	Text

User Guide

Job Search



GOV.UK Applicant Selection

Login

Job Search

Home

Viewing jobs

Job ID

Filter

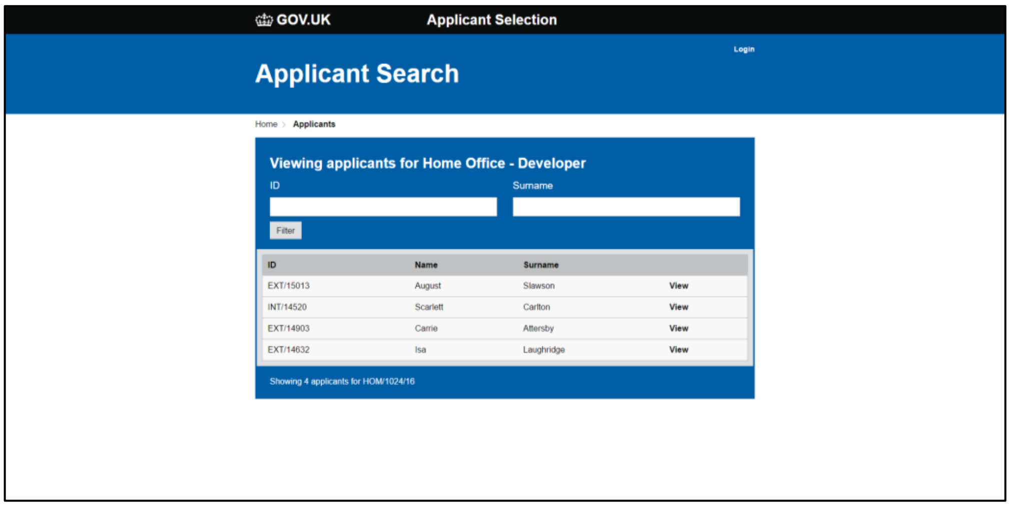
ID	Job	Closing date	
HOM/1024/16	Home Office - Developer	16/09/16	View Applicants
HOM/1025/16	Home Office - Junior Developer x 4	09/10/16	View Applicants
HOM/1026/16	Home Office - Senior Developer	23/09/16	View Applicants

Showing 3 jobs

The initial landing page of the web application displays a list of jobs, these can be filtered by ID which can be a full or partial query. This allows the user more flexibility, they could for example search the initial letters of the ID, specifying a particular organisation.

The user can click the 'View Applicants' link to proceed.

Applicant Search



GOV.UK Applicant Selection

Login

Applicant Search

Home > Applicants

Viewing applicants for Home Office - Developer

ID

Surname

Filter

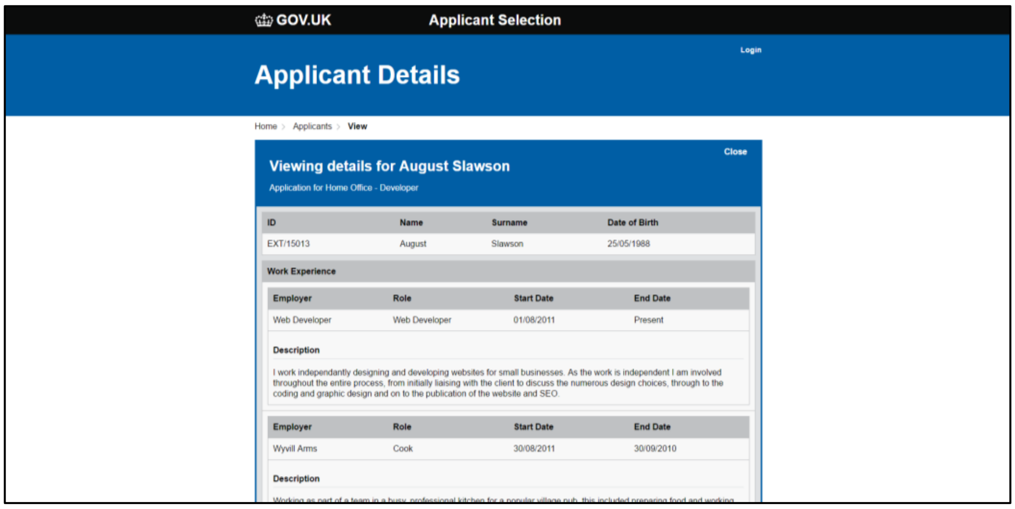
ID	Name	Surname	
EXT/15013	August	Slawson	View
INT/14520	Scarlett	Carlton	View
EXT/14903	Carrie	Attersby	View
EXT/14632	Isa	Laughridge	View

Showing 4 applicants for HOM/1024/16

This page displays a list of applicants for the previously selected job, displaying the ID, Name and Surname. The user can filter the results to find a specific applicant, much like the previous page the user can filter by a partial query.

The use can click the 'View' link to proceed to the next stage.

Applicant Viewer


GOV.UK Applicant Selection

Login

Applicant Details

Home › Applicants › View

Close

Viewing details for August Slawson

Application for Home Office - Developer

ID	Name	Surname	Date of Birth
EXT/15013	August	Slawson	25/05/1988

Work Experience

Employer	Role	Start Date	End Date
Web Developer	Web Developer	01/08/2011	Present

Description

I work independantly designing and developing websites for small businesses. As the work is independent I am involved throughout the entire process, from initially liaising with the client to discuss the numerous design choices, through to the coding and graphic design and on to the publication of the website and SEO.

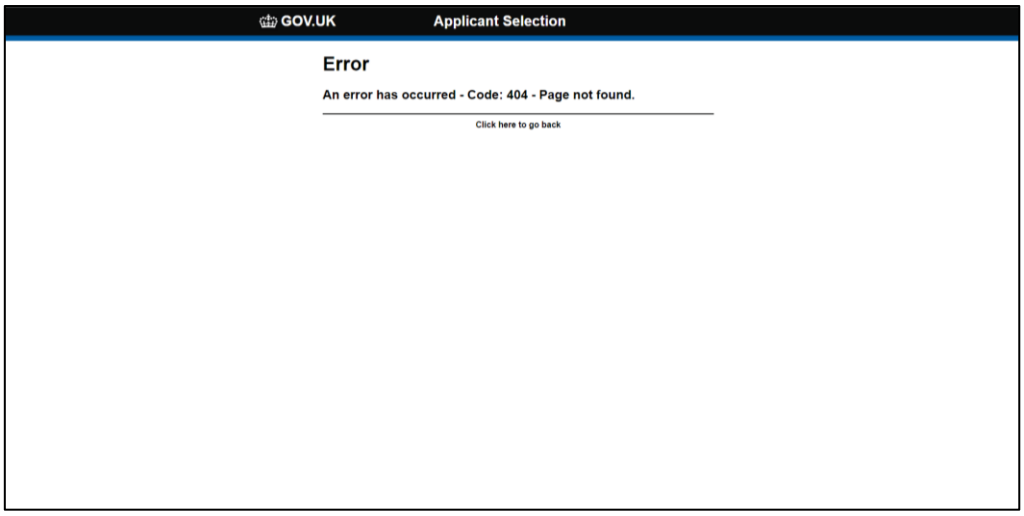
Employer	Role	Start Date	End Date
Wyvill Arms	Cook	30/08/2011	30/09/2010

Description

Working as part of a team in a busy, professional kitchen for a popular village pub, this included preparing food and working

This page displays more detail on the selected applicant, this includes Date of Birth and Employment History. The user can close this page and return to the previous page by clicking the top right of the window or they can go to any previous stage of the process by using the navigational below the header.

Error Pages


GOV.UK Applicant Selection

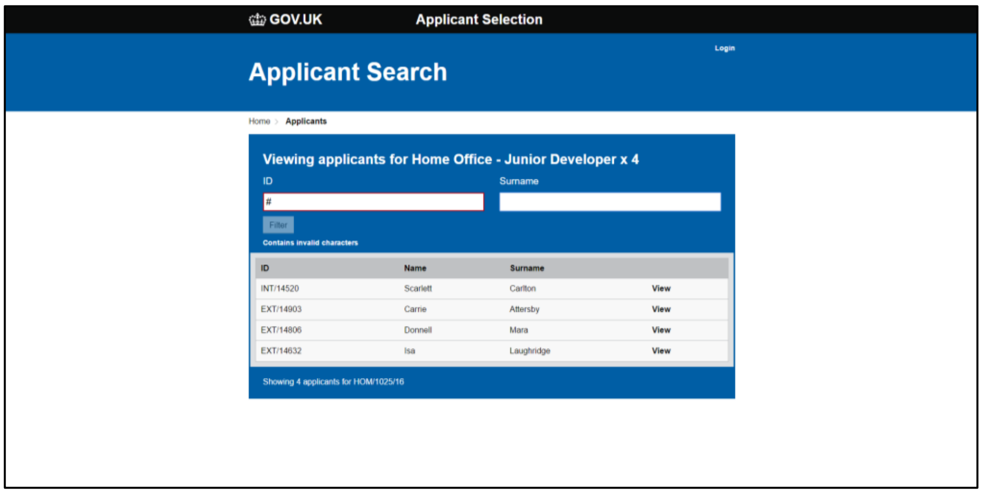
Error

An error has occurred - Code: 404 - Page not found.

Click here to go back

In the event that the server fails to render a page, due to an internal server error, database failure, or bad URL, the server handles the error and provides a dynamically generated error page. This page provides a description of the error and an error code, it also provides a link back to the previous page.

Data Validation


GOV.UK Applicant Selection

Login

Applicant Search

Home > Applicants

Viewing applicants for Home Office - Junior Developer x 4

ID: #

Surname:

Filter

Contains invalid characters

ID	Name	Surname	
INT/14520	Scarlett	Carlton	View
EXT/14903	Carrie	Attersby	View
EXT/14806	Donnell	Mara	View
EXT/14632	Isa	Laughridge	View

Showing 4 applicants for HOM/1025/16

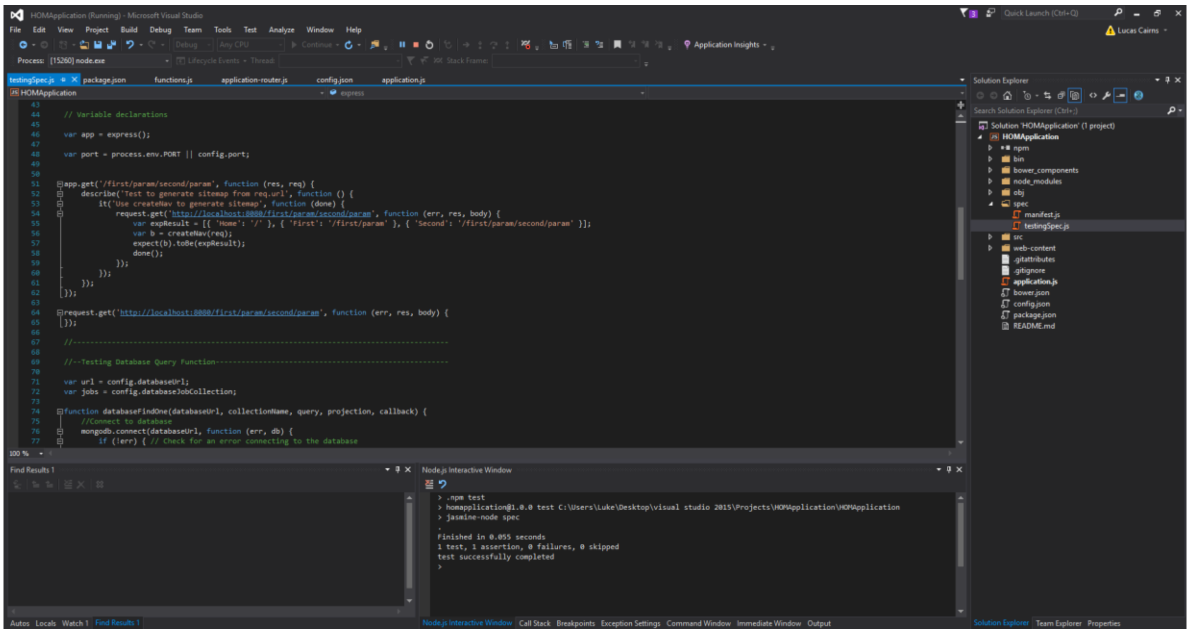
JQuery has been used to create dynamic form validation, the fields are highlighted whilst typing if the user exceeds the character limit for the specific field or enters an illegal character. The form then prevents submission of the invalid data by disabling the button and provides a message with detail on the invalid input.

Testing

Jasmine and Request have been used to run tests on a number of functions, two of which I have left in the project, these can be located in the spec folder and can be started through the NodeJs console using the command.

```
npm test
```

The test runs through both functions, one is a function that takes a URL and generates the JSON objects used to populate the navigation menu at the top of the page and the other function is used to connect with and query the database. Both functions were integral to the project and were required to be tested and bug free.



```
// Variable declarations

var app = express();

var port = process.env.PORT || config.port;


app.get('/first/param/second/param', function (res, req) {
    describe('Test to generate sitemap from req.url', function () {
        it('Use createNav to generate sitemap', function (done) {
            request.get('http://localhost:8080/first/param/second/param', function (err, res, body) {
                var expResult = [{ 'Home': '/' }, { 'First': '/first/param' }, { 'Second': '/first/param/second/param' }];
                var b = createNav(req);
                expect(b).toBe(expResult);
                done();
            });
        });
    });
});

request.get('http://localhost:8080/first/param/second/param', function (err, res, body) {
});

//---------------------------------------------------------------------------------

//--Testing Database Query Function-------------------------------------------------

var url = config.databaseUrl;
var jobs = config.databaseJobCollection;

function databaseFindOne(databaseUrl, collectionName, query, projection, callback) {
    //Connect to database
    mongodb.connect(databaseUrl, function (err, db) {
        if (!err) { // Check for an error connecting to the database
```

```
> .npm test
> homapplication@1.0.0 test C:\Users\Luke\Desktop\visual studio 2015\Projects\HOMApplication\HOMApplication
> jasmine-node spec
.
Finished in 0.055 seconds
1 test, 1 assertion, 0 failures, 0 skipped
test successfully completed
>
```