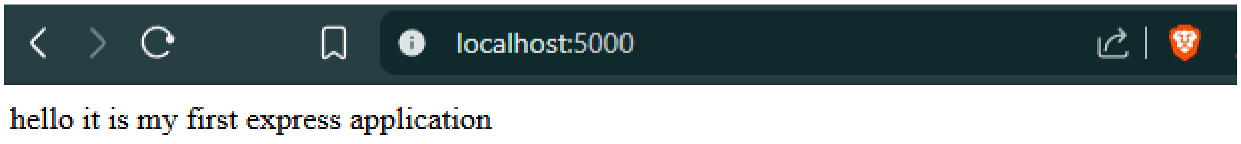
Cover Page

Reflective Report – CN5006 Week 4: Building a REST API with Node.js and Express

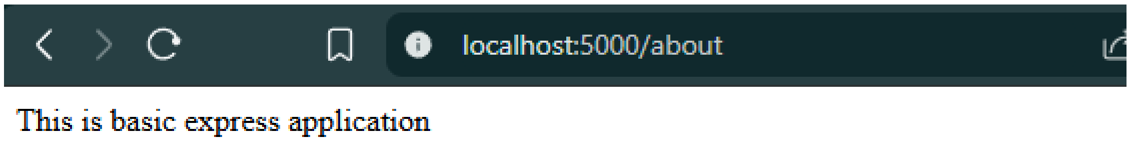
The lab of this week aimed at creating and testing an elementary RESTful API with the help of Node.js and Express framework. The challenge helped me to get acquainted with how to install a basic web server, implement a series of routes, read and serve data in the form of a JSON file and process POST requests with the assistance of a body-parser middleware. This task allowed me to reinforce my knowledge about how server-side JavaScript can handle HTTP requests and responses as well as how APIs can be used to help in structured communication between the client and the server.

The initial practice consisted of developing a simple Express application that served on port 5000 and responded with a basic message of hello with the help of the root route. This made me realize what an Express app is based on, such as the necessity to import fundamental modules like express, fs and body-parser. I was able to open the app in the browser with the help of the address of the local development server, which was http://localhost:5000, which made me think that the local development server was robust. This small, but important, lesson comprised how routing is carried out in Express and how to respond using text by using the res.send method.



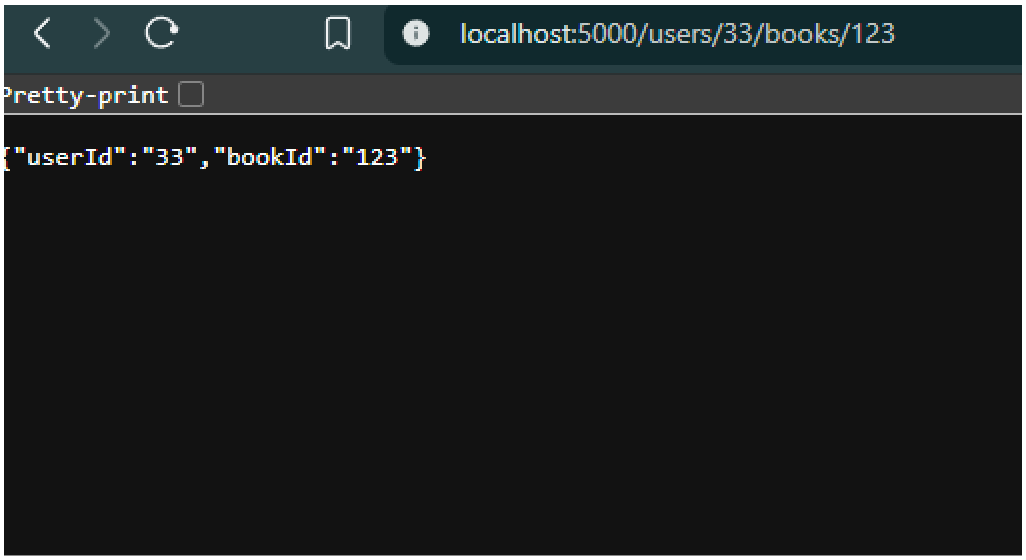
hello it is my first express application

The second practice brought in the dynamic routing and the URL parameters. Defining /about and /users/:userId/books/:bookId, I noted the way parameters are obtained by Express via the req.params. As an example, a query of /users/33/books/123 displayed a JSON object with both values. This activity illustrated how Express applications can develop malleable endpoints that get altered to various input parameters. It further taught me the importance of organizing and using syntax when making sure that there is no route conflict or syndrome in the bigger projects.



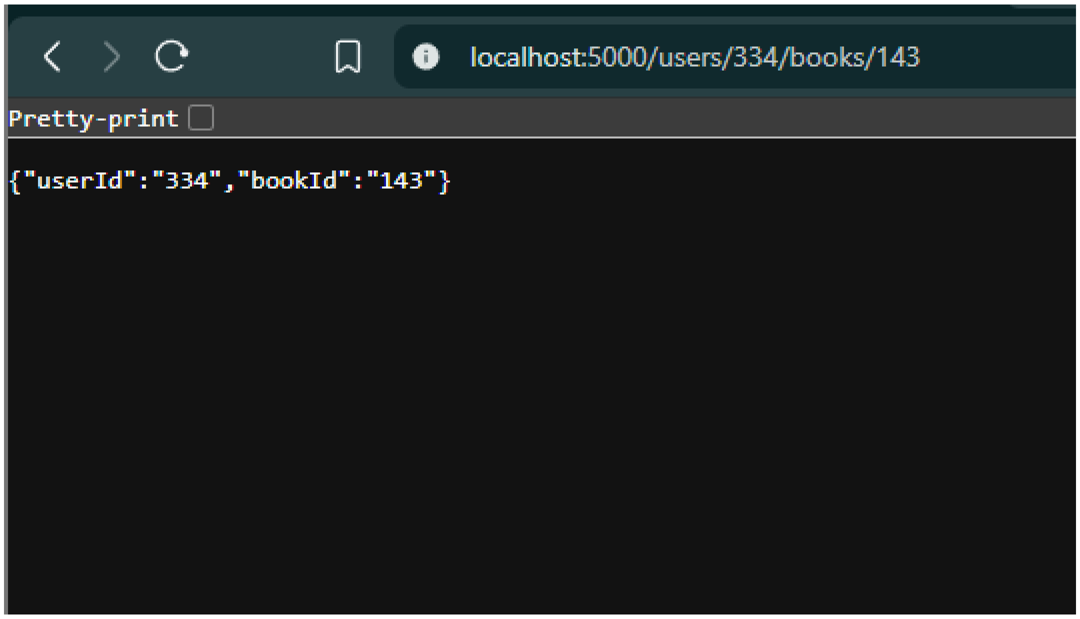
localhost:5000/about

```
This is basic express application
```



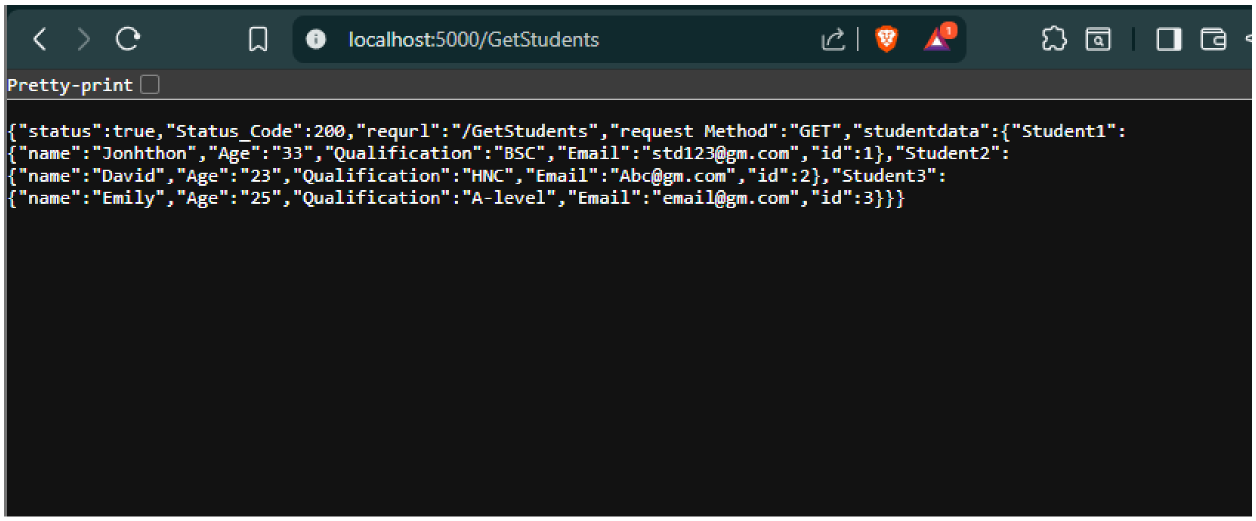
localhost:5000/users/33/books/123

```
{"userId":"33","bookId":"123"}
```

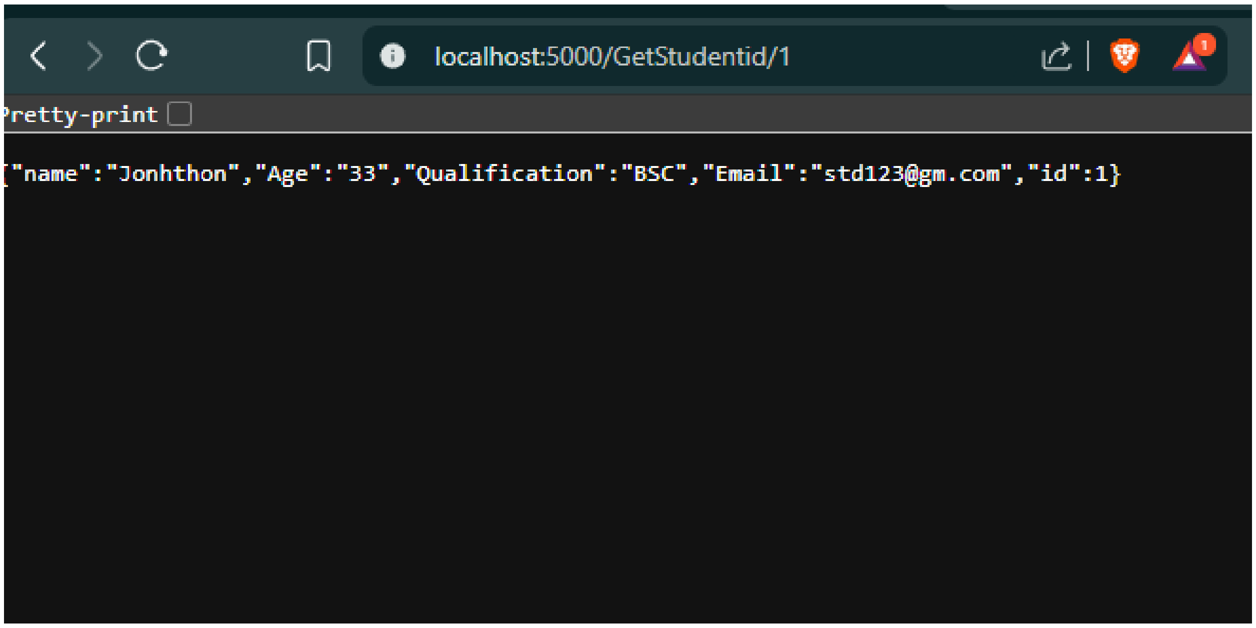


```
{"userId":"334","bookId":"143"}
```

The contents of the JSON file were read and parsed in the Express server by using Node built-in files module (fs). In Exercise 3, I generated a JSON file (Student.json) and read it using the built-in module (fs) within the Express server. With the help of specifying routes like /GetStudents and /GetStudentid/:id, I managed to get a list of all the student records or a particular record by using the parameter. The assignment was a good exercise in asynchronous file processing and parsing of JSON. I also knew how the use of a specific feature named as __dirname gives a built-in result of an absolute path of the working directory, which requires no extra manual manipulations of file-path parameters and enhances the portable character of the code in various settings.

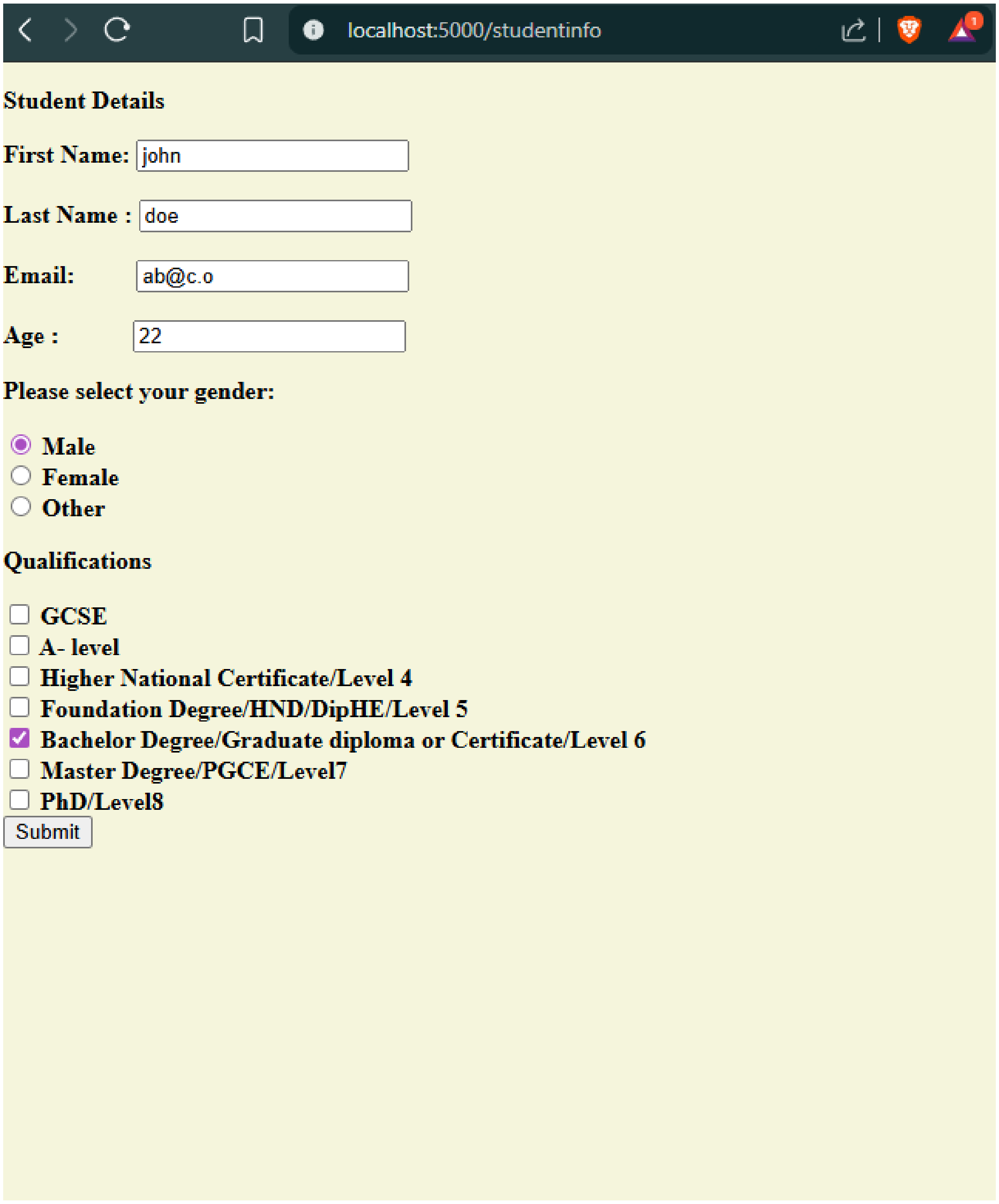


```
{"status":true,"Status_Code":200,"requrl":"/GetStudents","request Method":"GET","studentdata":{"Student1":
{"name":"Jonhthon","Age":"33","Qualification":"BSC","Email":"std123@gm.com","id":1},"Student2":
{"name":"David","Age":"23","Qualification":"HNC","Email":"Abc@gm.com","id":2},"Student3":
{"name":"Emily","Age":"25","Qualification":"A-level","Email":"email@gm.com","id":3}}}
```



```
{"name":"Jonhthon","Age":"33","Qualification":"BSC","Email":"std123@gm.com","id":1}
```

The last section of the lab was the introduction of POST method via an HTML form (StudentInfo.html). I used body-parser middleware that helped me capture the input of the user in the form and sent it back in the form of a JSON. In the case of dummy data that I entered, e.g. John Doe, 22 years, B.Sc, the answer was a structured JSON object that validates that the server received the POST request. This assisted me in linking the front end data submission with back-end API processing, the fundamental component of full-stack development.



localhost:5000/studentinfo

Student Details

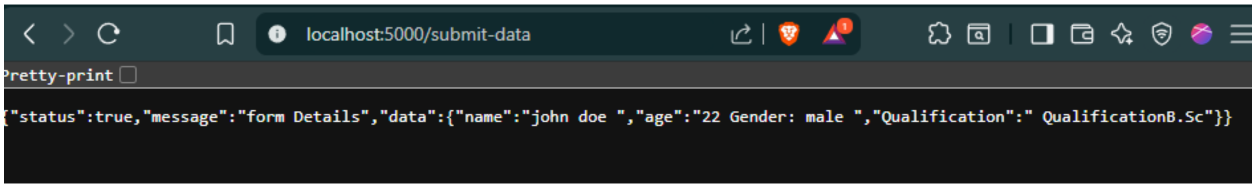
First Name:	john
Last Name :	doe
Email:	ab@c.o
Age :	22

Please select your gender:

- [✓ **Male**]
- **Female**
- **Other**

Qualifications

- **GCSE**
- **A- level**
- **Higher National Certificate/Level 4**
- **Foundation Degree/HND/DipHE/Level 5**
- [✓ **Bachelor Degree/Graduate diploma or Certificate/Level 6**]
- **Master Degree/PGCE/Level7**
- **PhD/Level8**



localhost:5000/submit-data

```
{"status":true,"message":"form Details","data":{"name":"john doe ","age":"22 Gender: male ","Qualification":" QualificationB.Sc"}}
```

In general, this lab enabled me to gain more knowledge on the REST architecture, Express routing, middleware application and file-system integration. I also got to know the logical flow of the requests made by client and the response the server gives. Throughout the experimentation and debugging, I was in a position to be able to read message in the console and browsers in order to diagnose the problems as fast as possible and so as to make sure that my API is functioning as intended. This practical experience has made me more confident in creating Node.js applications and more appreciative of how the concept of a RESTful API allows creating a scalable, efficient, and interactive communication over the web. I am now better equipped to create and implement basic web services and to build on those skills in the future when working on larger projects that require more complex frameworks or databases.