

web development report

by Salah Mohamed

Submission date: 15-May-2025 07:44PM (UTC+0100)

Submission ID: 259431892

File name: 58880_Salah_Mohamed_web_development_report_661685_636403852.docx (615.32K)

Word count: 1707

Character count: 9832

Web Development assignment report

NODE.JS, REACT.JS AND RNPS DOCUMENTATION
SALAH MOHAMED 1803148

Table of Contents

Introduction	2
Client-side and Server-side Scripting.....	3
Client-side scripting.....	3
Server-side scripting	3
Node.js framework	5
Documentation	7
User Manual.....	7
Homepage:.....	7
Add new plate:.....	7
View plates:	8
View customers	8
Register customers.....	8
Node.js and React Code.....	9
Testing.....	10
References.....	14

Introduction

In this report, the roles of client-side and server-side scripting in developing interactive and data-driven web applications are examined. Additionally, it explores Node.js as a runtime environment. The report's second half details the license plate management system, including its main components, structure, and testing procedure to show how these technologies can be used in real-world scenarios.

Client-side and Server-side Scripting

In web development, client-side and server-side scripting are essential paradigms that play distinct roles in facilitating smooth interaction between the web server and the client. Creating dynamic, flexible, and secure web applications requires a knowledge of their key differences.

Client-side scripting

This happens in the client's browser and the primary objective is to improve the user experience and interface by creating visually appealing webpages. Scripting language like JavaScript is executed in the client computer that allows for interactive features such as the use of animation, real-time updates, validating forms without needing to communicate with the server [1]. In client-side scripting, the browser is used as the run-time environment for interpreting and running code locally.

The below code is a simple example of client-side scripting used to improve interactivity by updating the page dynamically without directly interacting with the server.

```
function greetUser() {
  const name = document.getElementById('name').value;
  document.getElementById('greeting').textContent = "Hello, " + name;
}

<input type="text" id="name" placeholder="Enter your name">
Double-click to expand. =>greetUser()">Submit</button>
<p id="greeting"></p>
```

Server-side scripting

This happens on the web server and is mainly used to process user input, retrieve data from the database, and create dynamic content. Popular server-side scripting languages include PHP, Python, Java, and Node.js. A run-time environment like Apache, Node.js is necessary for these scripts as well as database connectivity such as PostgreSQL [2].

This endpoint is an example of server-side scripting, in which a backend controller method is used by the server to process, validate, and store incoming data from the client in the database.

```
@PostMapping  
public LicencePlate register(@Valid @RequestBody LicencePlate plate) {  
    return service.register(plate);  
}
```

Overall, there are a numerous key difference between the two in terms of functionality, execution environment and performance. Client-side scripting depends on browser compatibility and runs directly in the user's browser which can improves response time as no interaction with the server and database is needed which can enhance user experience and lowers the processing load on server. The source code is left exposed which poses a significant security risk. Whereas server-side scripting operates in web servers and remains hidden from the end user and produce preprocesses output, which is great for data security, but it increases processing load on the server as communication is required with the server. Also, it is quite important for managing database interaction, accessing files and providing an adaptable response depending on the user and access privileges [3].

To summarise, using server-side scripting can help in maintaining applicating logic and processing data whereas client-side scripting can be used improve the user experience and ensuring responsive content. Overall, both are important and serve as the foundation for most modern web development.

Node.js framework

Node.js is well known open-source runtime environment that allows JavaScript code to be executed outside of a web browser. It was first introduced by Ryan Dahl back in 2009 and it is built on top of Chrome's V8 JavaScript engine and is known for its flexible event driven features and autonomous architecture.

By enabling a language that was initially used for client-side to be used for server-side programming as well marked a dramatic change in the paradigm of web development. Over the past years Node.js became popular across application stacks due to its high performance, scalability and consistency.

One of its main advantages its asynchronous and event driven architecture which enables low resources consumption with a high concurrency. Node.js can easily manage numerous client requests using a single threaded event loop, in contrast with conventional multi-threaded server-side environments. Adopting this technique enables input and output operations such as file access and network communication to be handled effectively without disrupting the executing of other tasks which makes Node.js a great choice for applications like Apis and other lightweight web services that need to be responsive [4].

Using a single programming language throughout the application is a plus as JavaScript tools can be used for both client side and server side to streamline the application development process and significantly improve front-end and back-end team collaboration. Also, a large environment of reusable modules is available through NPM which can be used to substantially cut down on development time by giving different solution for common programming needs [4].

Furthermore, it improves loading time as it provides caching modules which support web applications to handle requests in simpler way and transfer it to in-application memory, which may lower loading time significantly and improve user experience.

Overall, using Node.js provides numerous benefits, like scalability, high performance, cost effective using full-stack JavaScript development. it is easy to learn and adapt, and it supports powerful data processing, concurrent request handling, and faster loading times and cross-platform development compatibility [5].

Node.js has drawbacks despite its advantages. While Node.js is quite perfect for input and output operations, but it Its not quite suitable for CPU intensive tasks or data processing tasks because these can block the event loop and cause lots of problems with performance. Although there are potential solutions like worker threads and clustering, they frequently come with additional complexity and possible performance drawbacks [6].

Another significant drawback is managing security and dependency as lots of Node.js applications usually depend on third party modules due to the fast NPM expansion. Some researchers have indicated that weaknesses in widely used NPM packages may jeopardise the applications security. For ensuring secure development it is quite important to keep packages constantly up to date, audit dependencies and avoiding complicated dependency chains [6].

Also, handling asynchronous code as this may bring up issues like callback hell, challenging debugging and subtle concurrency bugs even though modern syntax such as promises and `async` and `await` made the code a lot easier to read. To ensure proper handling of flow control and error handling developers much be knowledgeable about asynchronous design pattern [6].

To summarise, Node.js is a tool for creating applications that are rapid and scalable. Performance and development efficiency are improved by its unified and async design. Its async complexity, single-threaded architecture, and dependency management issues, however, call for careful thought. Node.js offers a contemporary and effective framework for a variety of web-based use cases; however, as with any technology, its application should be tailored to the particular requirements and environment of each application.

Documentation

User Manual

Firstly, to properly kick start the application ensure the following:

1. PgAdmin is running in the background and database is active
2. Run the backend Spring Boot application
3. Front end run npm install to ensure dependencies are up to date, then run npm start to start the React application. This will open the browser on <http://localhost:8081>

Homepage:

The homepage presents 2 options admin and customer view with various buttons as shown below.

Driver and Vehicle Licensing Agency

Get Your Personalised Plate

Admin View	Customer View
Add New Plate	View Plates
View Plates	Register Customer
View Customers	

Add new plate:

Selecting add new plate will forward you to this page fill all fields in the form. if the plate already exists or the plate matches a list of already defined inappropriate words, it will be rejects it.

Admin - Add Licence Plate

Plate Text:	
Price (£):	
Add Plate	

View plates:

This section shows existing plates and allows searching for plates via search bar, filter list by price, availability and already purchased and buy a plate.

Find your Personalised Registration

Filter by: None

Plate	Price (£)	Status	Purchase Date	Owner	Action
CCC	£150	Available	-	-	<button>Buy</button>

Filter by

✓ None

Price: Low to High

Price: High to Low

Available Only

Purchased Only

Plate

CCC

View customers

See all registered customers in the system.

Registered Owners

Name	Email	Driving Licence	Date of Birth	Address	Phone
Mike	mike@utsd.com	MK1234567MD	2025-05-08	1 Iq building	0123456789

Register customers

Customer registration is for registering new customer that can later buy a plate of their choice. Fill all required fields appropriately to successfully register in the system then click register button to complete registration.

Owner Registration

Full Name:

Mike

Email:

mike@utsd.com

Driving License Number:

MK1234567MC

Date of Birth:

08 / 05 / 2025

Address:

1 Iq building

Phone Number:

0123456789

Register

Node.js and React Code

The system frontend is created using React.js, while the development server is run using npm and packages are managed using Node.js as the runtime environment. The application is launched on <http://localhost:8081> thanks to Node.js and npm, which manage dependencies like npm install and local execution like npm start.

Index.js, which renders the root component and establishes the application's foundation, launches the application. HomePage.js, the primary landing page, has two distinct sections: Admin View and Customer View, each with buttons to access its own features.

Important elements consist of:

- Add plates: Allows the registration of new plate data, like price and text.
- Register customer: Allows new customers to sign up by inputting their personal data and submitting the form.
- View customers: Allows admins to see list of registered customers.
- View plates: Provides plates price and status filters for all available and purchased plates.
- Users can search and sort plates dynamically with the help of search bar and filter dropdown list.

The useState and useEffect hooks in React are used to handle side effects and dynamic data. The application updates the user interface in real time while retrieving license plate data from the backend API. To show whether plates are available or can be purchased, conditional rendering is utilised.

Testing

Adding new customer.

Owner Registration

Full Name:

Email:

Driving License Number:

Date of Birth:

Address:

Phone Number:

Register

Successful view customer.

Registered Owners

Name	Email	Driving Licence	Date of Birth	Address	Phone
Mike	mike@utsd.com	MK1234567MD	2025-05-08	1 lq building	0123456789

Add plate.

Admin - Add Licence Plate

Plate Text:

Price (£):

Add Plate

View plate

Find your Personalised Registration

Filter by: Price: High to Low

Plate	Price (£)	Status	Purchase Date	Owner	Action
M1KE	£300	Available	-	-	Buy

Buy plate

Find your Personalised Registration

Filter by: Price: High to Low

Plate	Price (£)	Status	Purchase Date	Owner	Action
MIKE	£300	Available	-	-	<button>Buy</button>

Plate assigned to mike

Plate	Price (£)	Status	Purchase Date	Owner	Action
MIKE	£300	Purchased	2025-05-15	Mike	

Search

Filter by: None

Plate	Price (£)	Status
CCC	£150	Available

Filter by available

Filter by: Available Only

Plate	Price (£)	Status	Purchase Date	Owner	Action
CCC	£150	Available	-	-	<button>Buy</button>
TEST	£100	Available	-	-	<button>Buy</button>
TEST2	£300	Available	-	-	<button>Buy</button>
NATHAN	£800	Available	-	-	<button>Buy</button>

Filter price low to high

Filter by: Price: Low to High ▾

Plate	Price (£)	Status	Purchase Date
TEST	£100	Available	-
CCC	£150	Available	-
MIKE	£300	Purchased	2025-05-15
TEST2	£300	Available	-
NATHAN	£800	Available	-

Filter by purchased.

Filter by: Purchased Only ▾

Plate	Price (£)	Status	Purchase Date	Owner
MIKE	£300	Purchased	2025-05-15	Mike

Validation.

 Please fill in this field.

Plate already exists

Plate Text:

Price (£):

Owner Registration

Full Name:	Mike
Email:	mike@utsd.com
Driving License Number:	MK1234567MD
Date of Birth:	08 / 05 / 2025
Address:	1 lq building
Phone Number:	0123456789

Error registering owner. Please try again.

References

- [1] J. Van Riet, "Client-side Performance of Web-based Applications: the State of the Art." Available: https://jaspervanriet.nl/assets/literature_study.pdf
- [2] X. Li and Y. Xue, "A survey on server-side approaches to securing web applications," *ACM Computing Surveys*, vol. 46, no. 4, pp. 1–29, Mar. 2014, doi: <https://doi.org/10.1145/2541315>.
- [3] T. Fadhilah Iskandar, M. Lubis, T. Fabrianti Kusumasari, and A. Ridho Lubis, "Comparison between client-side and server-side rendering in the web development," *IOP Conference Series: Materials Science and Engineering*, vol. 801, p. 012136, Jun. 2020, doi: <https://doi.org/10.1088/1757-899x/801/1/012136>.
- [4] B. Basumatary and N. Agnihotri, "Benefits and Challenges of Using NodeJS," *International Journal of Innovative Research in Computer Science & Technology*, vol. 10, no. 3, pp. 67–70, May 2022, doi: <https://doi.org/10.55524/ijircst.2022.10.3.13>.
- [5] N. Chhetri, "A Comparative Analysis of Node.js (Server-Side JavaScript)," *Culminating Projects in Computer Science and Information Technology*, Mar. 2016, Accessed: Apr. 02, 2024. [Online]. Available: https://repository.stcloudstate.edu/csit_etds/5/
- [6] H. Shah and T. R. Soomro, "Node.js Challenges in Implementation," *Global Journal of Computer Science and Technology*, Jun. 2017.

web development report

GRADEMARK REPORT

FINAL GRADE

60/100

GENERAL COMMENTS

Server/Client Code (25): 17

Node.js (25): 21

SW Doc (50): 22

FEEDBACK

=====

I was anticipating much more discussion of the runtime environments and the differences between running on client vs server.

This is very brief but does touch on some of the key concepts.

Good discussion of node.js which covers the feature and benefits quite well. I would have liked a little more on why node.js has good performance.

There is a number of screen shots but I was expecting more discussion of the code and the testing to be provided.

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15
