

Suman Kharel

## **NOKIA 5G FIELD TEST WEB APPLICATION**

Full Stack JavaScript Application

# **NOKIA 5G FIELD TEST WEB APPLICATION**

Full Stack JavaScript Application

Suman Kharel  
Bachelor's Thesis  
Spring 2019  
Degree Programme in  
Information Technology  
Oulu University of Applied Sciences

## ABSTRACT

Oulu University of Applied Sciences  
Degree Programme in Information Technology

---

Author: Suman Kharel

Title of Bachelor's thesis: Nokia 5G Field Test Web Application

Supervisor: Jukka Jauhiainen

Term and year of completion: Spring 2019

Number of pages: 47

---

The purpose of this bachelor's thesis was to develop the full stack JavaScript Application for Nokia. Nokia is a Finnish multinational telecommunication, information technology and consumer electronic company headquartered in Espoo, Finland [1]. Nokia is among the forerunner of the development of the 5G network. Nokia's 5G test team perform field testing to verify that the radio network is performing as specified over the air environment. The team needed a web application to store log files obtained from the test and to visualize data for the effective measurement of the network.

The author developed the application using MERN (MongoDB, Express.js, React.js, Node.js) stack. A modern web application was developed fulfilling all the initial requirements set by the team. The application stores uploaded log files to a MongoDB database, visualizes log files onto a map, and in 3D.

---

Keywords:

Full stack JavaScript Applications, Web Applications, MongoDB, Express.js, ReactJS, Node.js, Redux

# CONTENTS

1	INTRODUCTION .....	7
2	CELLULAR NETWORK.....	8
2.1	1G (First Generation) .....	8
2.2	2G (Second Generation) .....	8
2.3	3G (Third Generation) .....	8
2.4	4G (Fourth Generation) .....	9
2.5	5G (Fifth Generation) .....	9
3	MODERN JAVASCRIPT WEB APPLICATIONS.....	11
3.1	Primary client-side web technologies .....	11
3.1.1	HTML .....	11
3.1.2	CSS .....	12
3.1.3	JavaScript .....	13
3.2	Multi Page Applications .....	14
3.3	Single Page Applications.....	15
3.4	Progressive Web Application.....	16
4	WEB DEVELOPMENTS TOOLS .....	18
4.1	Editors .....	18
4.2	Version control system .....	19
4.3	API development and test tools.....	22
5	NOKIA 5G TEST DASHBOARD .....	24
5.1	Requirements .....	24
5.1.1	Zip Folder requirements .....	24
5.1.2	Application requirements.....	25
5.2	Backend .....	25
5.2.1	Introduction to technologies used .....	26
5.2.2	Restful APIs .....	28
5.3	Frontend.....	30
5.3.1	Introduction to technologies used .....	30
5.3.2	Main Menu .....	33
5.3.3	Map View .....	37
5.3.4	Calendar View.....	39

5.3.5	Selected Files View.....	40
5.3.6	Drone 3D View.....	42
6	CONCLUSION.....	43
	REFERENCES.....	44

## VOCABULARY

WORD	DEFINITION
1G	First Generation
2G	Second Generation
3D	Three Dimension
3G	Third Generation
4G	Fourth Generation
5G	Fifth Generation
CDMA	Code Division Multiple Access
CPU	Central Processing Unit
CSS	Cascading Style Sheets is a stylesheet language used to describe the presentation of a document written in a mark-up language like HTML
CSS3	CSS3 is the latest evolution of Cascading Style Sheets language that extends CSS2
EDGE	Enhanced Data Rates for Global Evolution
Gbps	Gigabits per second, a unit of measurement of bandwidth
GHz	Gigahertz, a unit of measurement of frequency
GSM	Global System for Mobile
HTML	Hyper Text Mark-up Language describes and defines the content of a webpage along with the basic layout of the webpage
HTML5	HTML5 is the fifth and current major version of the HTML
JSON	JavaScript Object Notation
Kbps	Kilobits per second, a unit of measurement of bandwidth
KHz	Kilohertz, a unit of measurement of frequency
LTE	Long-Term Evolution
Mbps	Megabytes per second, a unit of measurement of bandwidth
MHz	Megahertz, a unit of measurement of frequency
UI	User Interface
UMTS	Universal Mobile Telecommunications System
URL	Uniform Resource Locator
WiFi	Wireless Fidelity

# 1 INTRODUCTION

Nokia is a Finnish multinational telecommunication and information technology company. It was founded in 1865 as a pulp mill and has succeeded in several business sectors over the years. Among many other industries such as cable, paper products, rubber tires and boots, Nokia have acquired success in mobile network industry. Nokia have contributed to the development of the GCS, 3G, LTE standards and currently focusing in the development of the 5G network. Nokia is best known for its phone, which once was the largest vendor in the world.

Established as a Finnish market focused company in 1865, Nokia expanded to Nordic and European market in the 1990s. From the mid-1190s onward, Nokia became a truly global company. [2]

Nokia is among the major companies that are focused in the development of the 5G network. 5G is the newest generation of radio system and the network architecture promising much faster data download and upload speeds, wider coverage and more stable connections. The faster data transmission rate is expected to unleash many opportunities in the future.

One of the integral parts of the 5G development is field testing. Field testing is performed to verify that the radio network is performing as specified over the air environment. The field test team verifies that the RF unit performance is according to the product specification.

The main aim of this thesis was to develop a web application for Nokia's 5G team. The 5G field test team needed a web application to store and visualize log files obtained from the field tests.

## **2 CELLULAR NETWORK**

A cellular network, also known as a mobile network, is a network of mobile base stations that provides coverage to be able to connect mobile phones to phone calls, text messages and data services. The base stations radiate electromagnetic waves and the mobile phones within that area can connect to the network.

The following section explains the evolution of the cellular network from 1G (First Generation) to 5G (Fifth Generation)

### **2.1 1G (First Generation)**

The first generation of a cellular network was launched commercially in Japan in 1970. 1G used an analogue radio signals to connect to the network. [3]

### **2.2 2G (Second Generation)**

The GSM (Global System for Mobile Communication) was introduced in the second generation of a mobile network. It enabled the users to send the advanced voice and information over the network. The second generation of network utilized Signaling, Mobile Stations Authentication to guarantee enhanced security and protection of phone calls. The development in innovation from 1G to 2G presented many services that are used today, such as SMS, data roaming, phone calls, call hold and billing based on services used. [3]

### **2.3 3G (Third Generation)**

The third generation of a mobile network was launched in 2001 with an aim to provide the significant voice and information capacity supporting many applications with increased data transmission at a lower cost. 3G assisted the fast-wide band internet access. The video call, navigational maps, emailing, mobile gaming, music and digital services were used in the 3G network. [3]



## 2.4 4G (Fourth Generation)

The fourth generation of a mobile network was launched in 2010 with an aim to introduce fast, cheap and quality internet with secured connection. The 4G technology builds upon what 3G offers but does everything at a much faster speed. The 4G internet connection is around ten times faster than 3G. [3]

## 2.5 5G (Fifth Generation)

5G is the newest generation of a commercial cellular network expected to be launched in early 2019. It is set to greatly increase the internet connectivity speeds. Beyond just speed improvements, 5G is expected to unleash massive IoT ecosystem where networks can serve communication needs for billions of connected devices, with the right trade-offs between speed, latency and cost. [3]

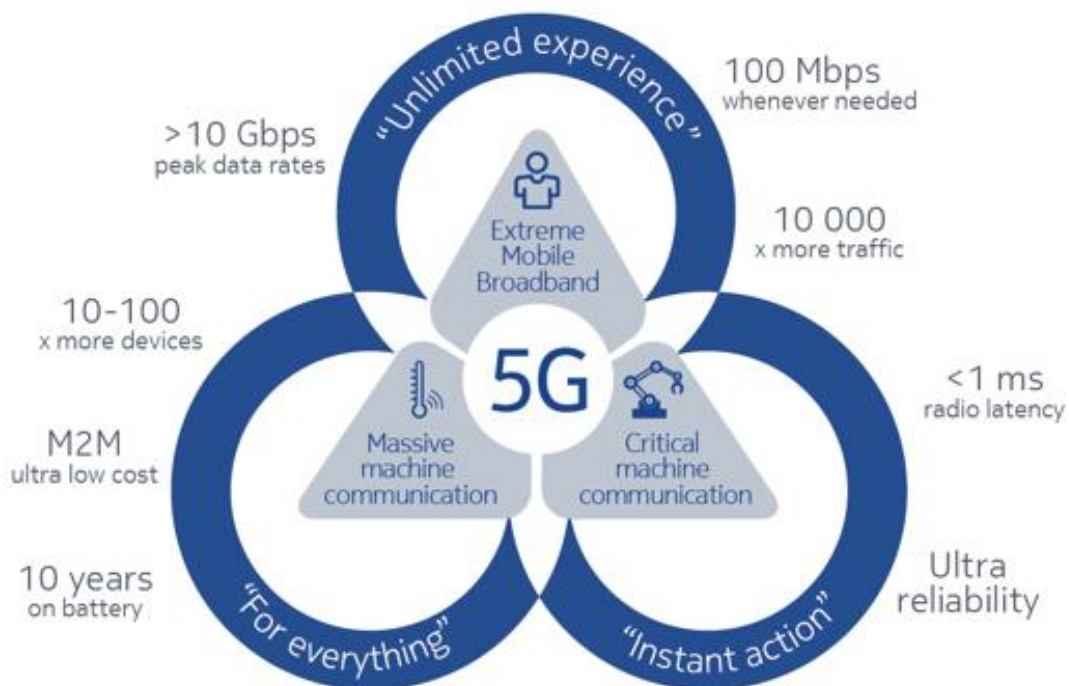


FIGURE 1. Targets of 5G [4]

As seen in the figure 1, 5G will power massive broadband applications and provide unlimited experience. Among many other use cases, 5G can be used in critical machine communications with ultra-reliability because of its low latency.

### 3 MODERN JAVASCRIPT WEB APPLICATIONS

A web application is a program that runs in the web server and can be accessed through a web browser. The common web applications include Facebook, Gmail, YouTube. [5]

Web development has come a long way from static websites to advanced web applications. The following section describes the common technologies used for web development, and the evolution of web applications.

#### 3.1 Primary client-side web technologies

The following section describes the primary client-side web development technologies used for developing web applications

##### 3.1.1 HTML

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. Its elements form the building blocks of pages, representing formatted text, images, form inputs, and other structures. When a browser makes a request to a URL, whether fetching a page or an application, the first thing that is returned is an HTML document. The returned HTML document may reference or include additional information about its look and layout in the form of CSS, or behavior in the form of JavaScript. [6]

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML document</title>
  </head>
  <body>
    <h1>Simple HTML document</h1>
  </body>
</html>
```

FIGURE 2. HTML document

A typical plain HTML document as seen in the Figure 2, shows that the document starts with the declaration of the version of HTML to be used. In this case, `<!DOCTYPE html>` declares this document to be HTML5. The `<html>` element is the root element of the HTML page. The `<head>` element contains the meta information about the document. The `<body>` element contains the visible part of the web page. The `<h1>` element defines a large heading, which is displayed on the web page as shown in the figure 3.



*FIGURE 3. Example document displayed on the browser*

### **3.1.2 CSS**

Cascading Style Sheets (CSS) is used to style the web page. CSS style sheets can be applied directly to an HTML element, defined separately on the same page, or defined in a separate file and referenced by the page. Styles cascade is based on how they are used to select a given HTML element. For instance, a style might apply to the entire document, but would be overridden by the style that applied to a particular element. Likewise, an element specific style would be overridden by a style that applied to a CSS class that was applied to the element, which in turn would be overridden by a style targeting a specific instance of that element. [6]

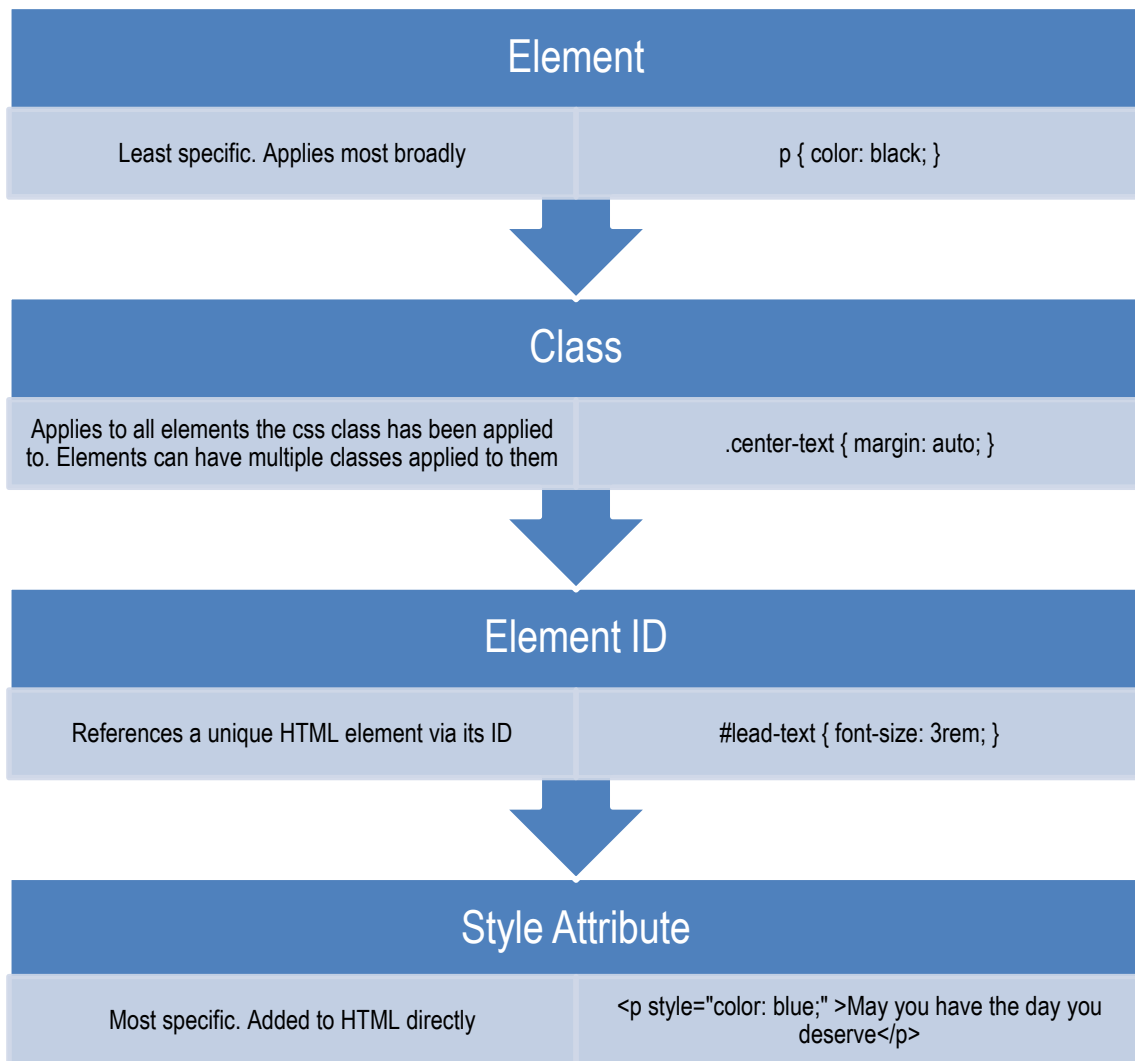


FIGURE 4. CSS specificity rules, in order from the least specific to the most specific [6]

### 3.1.3 JavaScript

JavaScript is an object-oriented programming language used to make the interactive web applications. JavaScript has evolved from a mere client-side language to a complex server-side language.

Client-side JavaScript is used to handle the browser interaction and its Document Object Model (DOM). For example, JavaScript enables the application to add a content in the web page when a button is clicked.

Server-side JavaScript, such as Node.js runs in the server and handles the custom requests made by the client-side of the application. Node.js enables an application to communicate with a database and perform the required actions. [7]

```
<!DOCTYPE html>
<html>
  <body>
    <h2>What can JS do in browser?</h2>
    <p id="demo">Click the button to change this paragraph.</p>
    <button type="button" onclick="changeParagraph()">Change Paragraph</button>

    <script>
      // change text content on button click
      function changeParagraph() {
        document.querySelector("#demo").innerHTML =
          "Wow! Can't believe you did it. ";
      }
    </script>
  </body>
</html>
```

FIGURE 5. JavaScript in the browser

The figure 5 shows the implementation of JavaScript inside the HTML document. The code enclosed inside the `<script>` tag adds the interactivity to the HTML page. When the user clicks on a Change Paragraph button, the text is updated as shown in the figure 6.



FIGURE 6. Implementation of JavaScript for interactivity in a web application

### 3.2 Multi Page Application

Multi Page Application (MPA) is a traditional kind of web app. As seen in the figure 7, every time the application needs to display data or submit data back to the server, it must request a new page, which is then rendered in the web browser. This approach is fine for simple applications but if the app becomes more complex with lots of data or an intricate interface, it may run into

performance issues. As seen in the figure 7, generating pages on a server, transferring them to the client, and rendering them into the browser takes time and can degrade the user experience. Additionally, every time the app calls the server, it triggers a page refresh in the browser. [8]



FIGURE 7. Multi page application lifecycle [9]

### 3.3 Single Page Applications

A Single Page Application (SPA) is an application that runs in the browser and loads the contents by dynamically rewriting the page when a user interacts with it. In a single page application, either all the required content is loaded during the initial load or the additional pages are dynamically loaded when the user interacts with the page. Location hash or HTML5 history API can be used to navigate between the pages without reloading the page. The interaction between the application and the server happens behind the scene as seen the figure 8. The single page applications can be developed using different techniques such as Ajax, WebSocket, JavaScript Frameworks. [8]

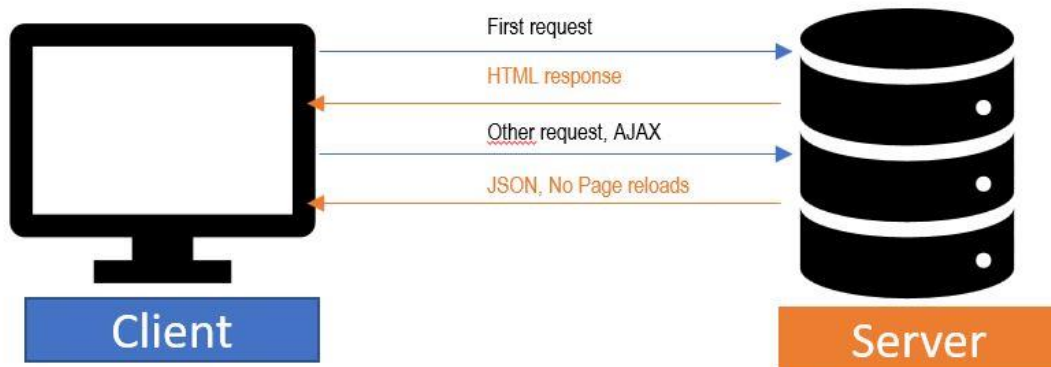


FIGURE 8. Single page application lifecycle [9]

### 3.4 Progressive Web Application

A Progressive Web App (PWA) is the next level of browser-based application. While SPA has already meant a giant leap forward, PWA is taking the web application one step closer to native applications.

A Progressive Web Application is an application that runs in the browser like regular web applications with additional features such as offline load, notifications, and access to device hardware. They use service workers, HTTPS, a manifest file and an app shell architecture to deliver native app experience to web applications. [10]

Progressive Web apps can be installed on the user's home screen. The characteristic is enabled by a manifest file that describes how an app is supposed to behave when run outside of the browser. As seen in the below example code, the web app manifest file is a simple JSON file with key-value pairs that configure the main aspects of the app.



```
{
  "short_name": "Dashboard",
  "name": "5G Test Dashboard",
  "icons": [
    {
      "src": "/images/nokia.png",
      "type": "image/png",
      "sizes": "192x192"
    },
    {
      "src": "/images/nokia.png",
      "type": "image/png",
      "sizes": "512x512"
    }
  ],
  "start_url": "/dashboard/?source=pwa",
  "background_color": "#3367D6",
  "display": "standalone",
  "scope": "/dashboard/",
  "theme_color": "#3367D6"
}
```

FIGURE 9. A complete manifest.json file for PWAs [11]

In the Figure 9, `short_name` is displayed on the user's home screen or while launching the application. The `name` property is used in the app install prompt. The `icons` property is an array of images objects that defines a set of icons for the browser to use when it is added to the user's home screen. The `start_url` includes the link to initial landing page of the application. The splash screen uses the `background_color` property while launching the application. The `standalone` value in a `display` property loads the application in a new browser window with hidden browser UI element such as the URL bar. The `scope` property defines the set of URLs that the browsers consider to be within the app, and is used to decide when the user has left the app. The `theme_color` is used in the tool bar and app's preview during task switches. [11]

## 4 WEB DEVELOPMENTS TOOLS

Web development tools have come a long way in just a few short years. Thanks to this progress, developers can harness the power of highly tested libraries to improve their workflow. With the number of web dev tools increasing almost daily, finding the best software to get the job done can sometimes feel daunting. Developers have many options to choose from. The following tools were used during the development of the application.

### 4.1 Editors

A code editor plays a vital role for web developers while working on big projects. Finding one that best fits the workflow and has all the latest features needed, can save a lot of time and significantly improve the code quality. Some of the free and popular code editors in the web development community as of 2018 are Visual Studio Code [12], Atom [13], Brackets [14] and Sublime Text [15].

Based on its excellent JavaScript support and helpful addons, Visual Studio Code was selected as a code editor for this project. The figure 10 shows the list of helpful vs code addons used while developing the application.

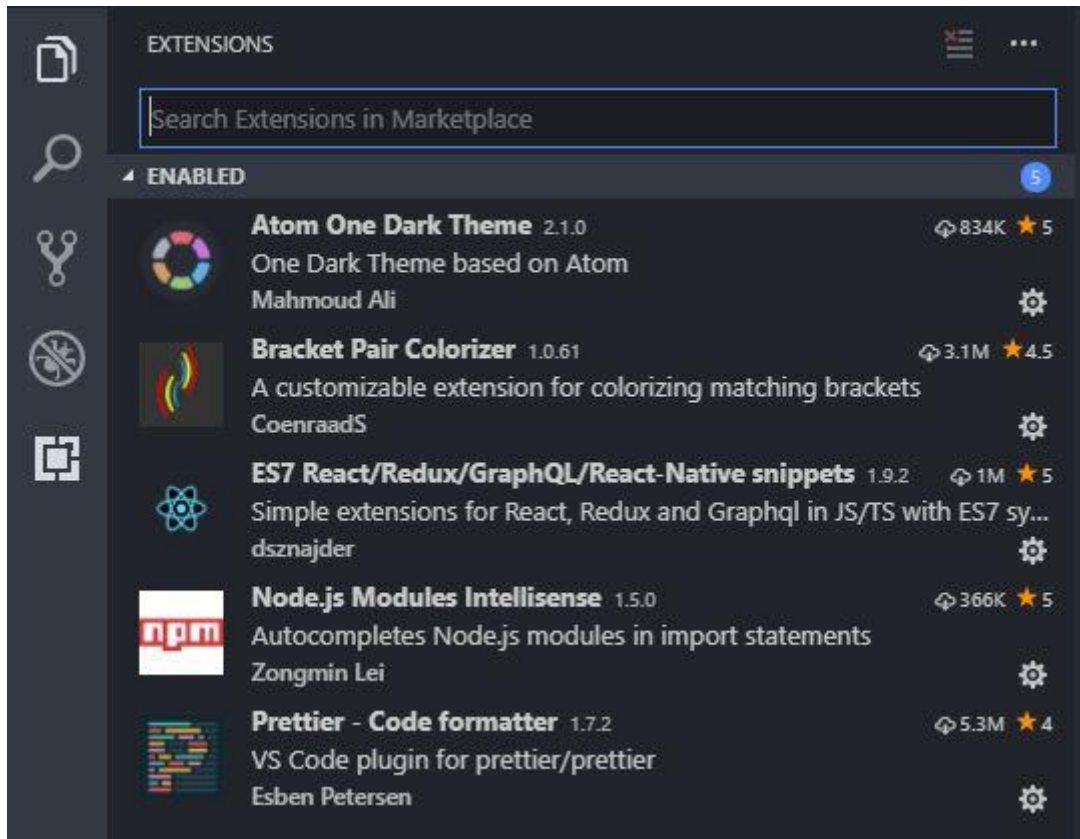
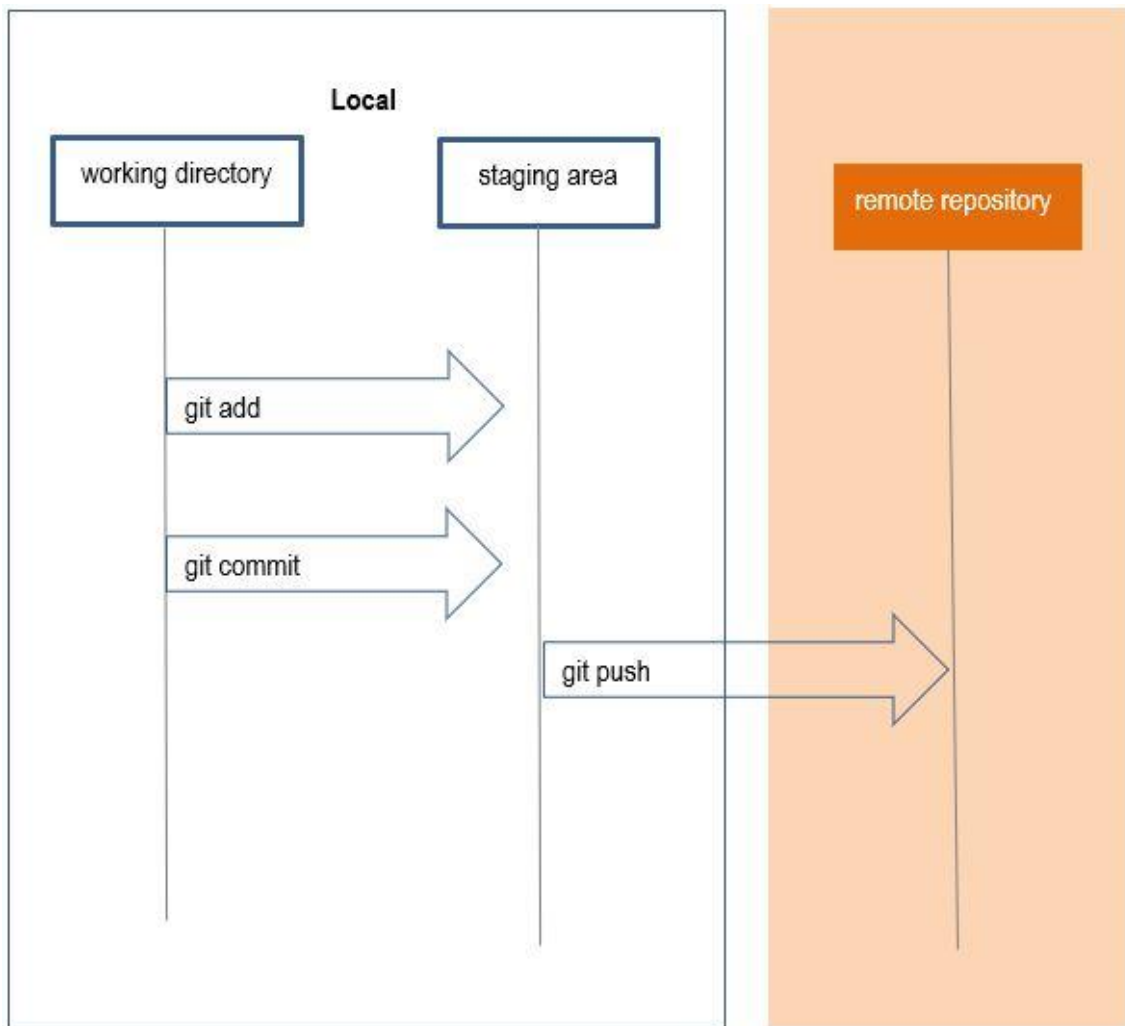


FIGURE 10. VS Code editor setup with addons

## 4.2 Version control system

A version control system is a software tool that helps software developer teams to manage changes to source code over time. Version control software keeps track of every modifications to the code. If need be, developers can revert or get access to earlier versions of code. Some of the popular version control systems are GIT [16], CVS [17], Mercurial [18] and Bazaar [19].

Nokia uses Gitlab as one of the web platforms for hosting source code of projects, so GIT was selected as a version control system for this project.



*FIGURE 11. Git workflow during the application development*

As seen in the figure 11, Git was used to publish the changes made in the local repository. The interaction to the remote repository is one way because the author was the only developer working in the project.

The workflow might be slightly different for a team working in the same project. Developers usually create their own branch and when the feature is completed, they merge it to the main branch. Once merged, other developers pull the changes from the main branch before committing their changes to the main branch.

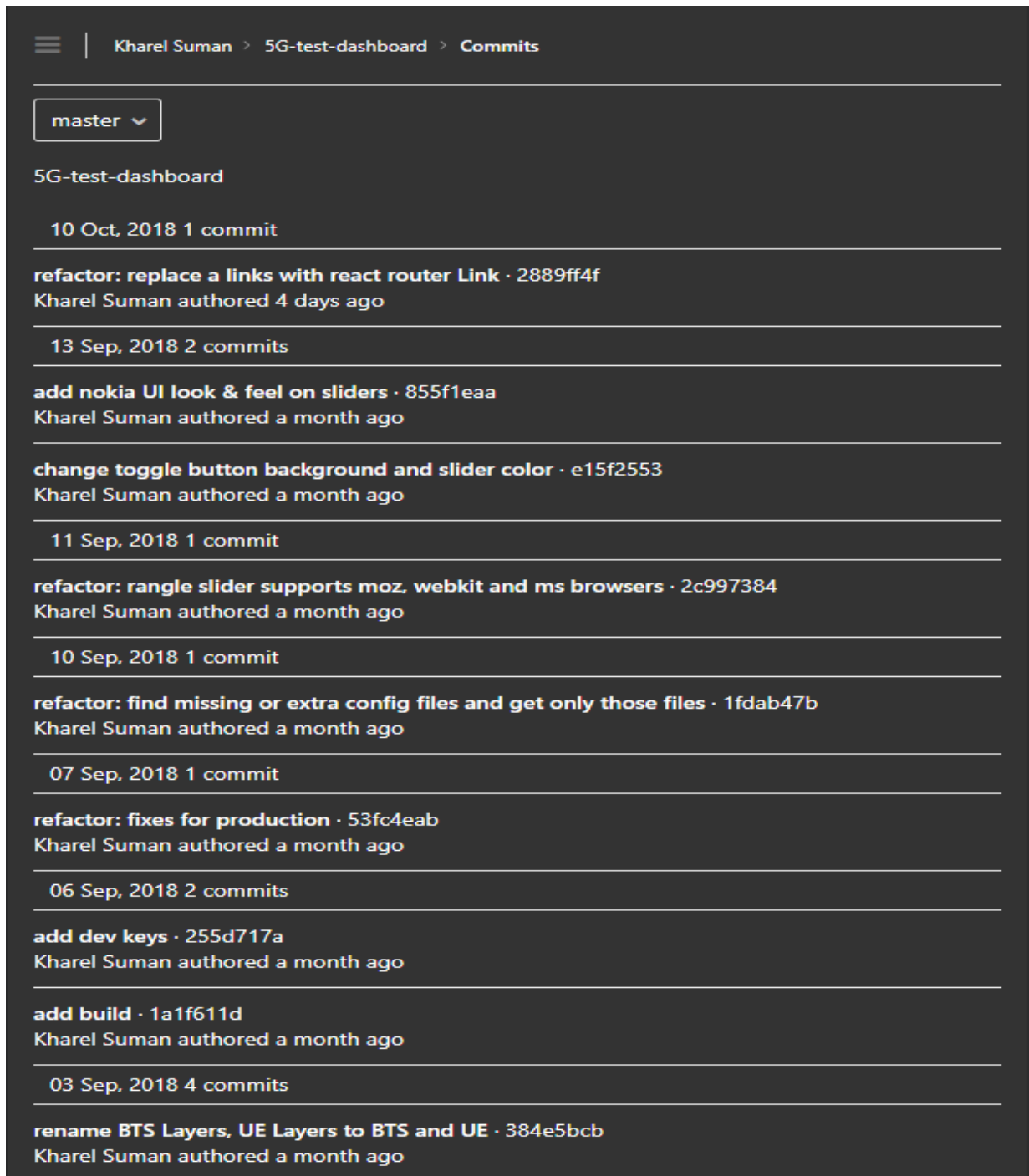


FIGURE 12. Git commits of 5G test dashboard [20]

As seen in the figure 12, frequent commits were made to publish the changes made in the local repository. Commit messages were written in a way that it could explain the purpose of added code or features onto the existing remote source code. The commit message that started with a refactor text implied that the source code was updated without modifying the application behavior.

### 4.3 API development and test tools

API is a piece of software that plugs one application directly into the data and services another by granting it access to the specific parts of software. APIs let two pieces of software communicate.

There are several tools available for API developments and testing. Some of the popular tools as of 2018 are Insomnia [21], REST-Assured [22], SoapUI [23], JMeter [24] and Postman [25].

Insomnia is a free cross-platform desktop application that simplifies the interaction with HTTP-based APIs. Insomnia combines an easy to use interface with advanced functionality like authentication helpers, code generation and environment variables [21]. Because of these reasons, Insomnia was selected as an API development and testing tool for this project. The figure 13 shows some of the APIs created under the 5G Test Dashboard folder.

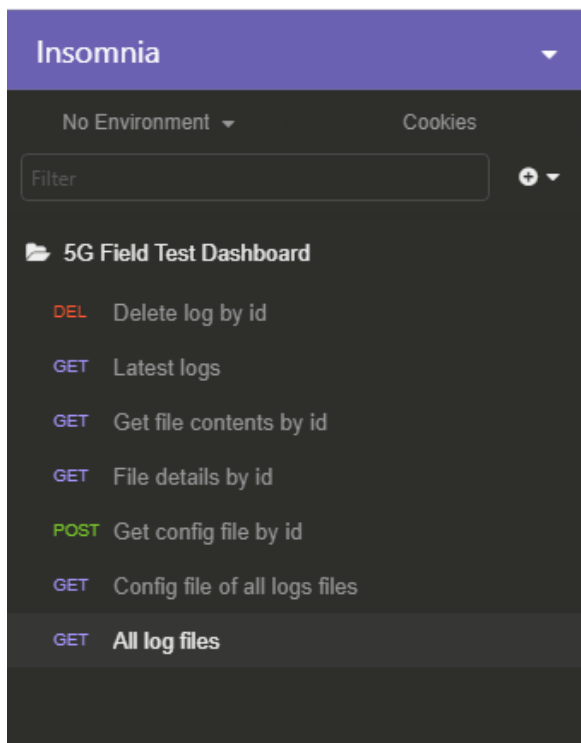


FIGURE 13. Application routes collection in insomnia

Developers can select any route from the collection and perform the HTTP request to verify that the route is performing as intended. As seen in the figure 14, the GET request to `api/logs/files` responded the list of uploaded log files with a status code 200.

200 OK

TIME 187 ms

SIZE 21.6 KB

```
Preview ▾ Header 7 Cookie Timeline
1 * [
2 * {
3   "_id": "5b962dd660b23eb2e4a9de04",
4   "length": 215451,
5   "chunkSize": 261120,
6   "uploadDate": "2018-09-10T08:39:51.565Z",
7   "md5": "6699433aad3cdfc46be7fac699901877",
8   "filename": "10.09.2018 - json_sample_package - BTS missing.zip",
9   "contentType": "application/x-zip-compressed"
10  },
11 * {
12   "_id": "5b962d26b889548f308b42e2",
13   "length": 215451,
14   "chunkSize": 261120,
15   "uploadDate": "2018-09-10T08:36:55.055Z",
16   "md5": "6699433aad3cdfc46be7fac699901877",
17   "filename": "10.09.2018 - json_sample_package - BTS missing.zip",
18   "contentType": "application/x-zip-compressed"
19  },
20 * {
21   "_id": "5b7a9d428386675e1c6c1f01",
22   "length": 286817,
23   "chunkSize": 261120,
24   "uploadDate": "2018-08-20T10:51:46.889Z",
25   "md5": "68876c1e92460951dae50e047cbe94aa",
26   "filename": "20.08.2018 - json_sample_package.zip",
27   "contentType": "application/x-zip-compressed"
28  },
29 * {
30   "_id": "5b7a9be12dd5c25a4431d77d",
31   "length": 286817,
32   "chunkSize": 261120,
33   "uploadDate": "2018-08-20T10:45:54.437Z",
34   "md5": "68876c1e92460951dae50e047cbe94aa",
35   "filename": "20.08.2018 - json_sample_package.zip",
36   "contentType": "application/x-zip-compressed"
37  },
38 * {
```

FIGURE 14. Testing APIs with insomnia

## 5 NOKIA 5G TEST DASHBOARD

A 5G test dashboard is a single page web application that visualizes the 5G field test data. It includes features like file validations, extensive filters, 2D and 3D visualization to enable the team to effectively measure field test data.

A zip folder (hereafter called as a log file) containing required information as mentioned below is uploaded to the database, which can later be selected to visualize data. A log file is the automatically produced and time-stamped data obtained from the test by Nokia's field test cars and drones.

### 5.1 Requirements

A dashboard requires an uploaded log file to visualize data. The log file is validated before uploading it to the database. The folder structure with required files and optional files are mentioned below.

#### 5.1.1 Zip Folder requirements

A zip folder consists of required and optional files. As seen in the figure 15, a sample folder has four subfolders and one JSON file.

While validating the zip file, the application checks if **testconfig.json** and **bts.json** exists. If not, the upload is rejected with an error message. If **ue.json** does not exist, warning dialogue is shown before uploading it to the database. The figure 15 shows the expected folder structure.



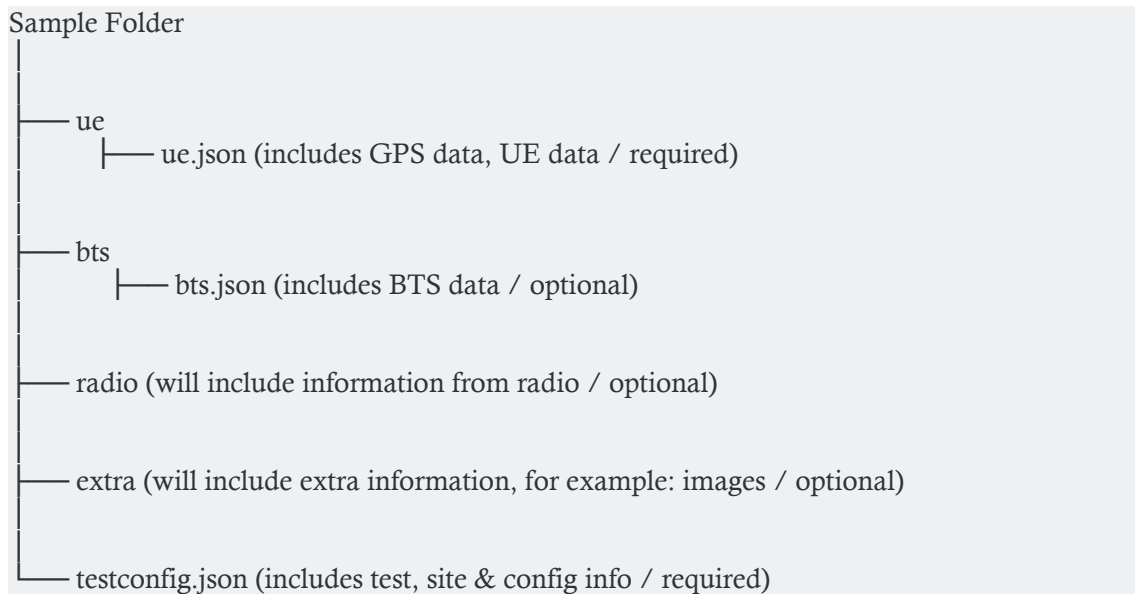


FIGURE 15. Sample folder structure

### 5.1.2 Application requirements

Several meetings were held with the Radio Frequency (RF) field test team during the application development period. The following ideas were discussed in the first RF test team meeting.

1. The RF test team needs a database to store log files obtained from each field test
2. The RF test team needs a modern web application to list stored log files
3. The Application design should have Nokia's UI look & feel
4. The Application should have the extensive filters to enable the team to find intended log files effortlessly
5. The Application should visualize the selected log file
6. The Application should allow to compare multiple log files

The full stack web application is composed of two broad parts: frontend and backend which are explained below.

## 5.2 Backend

Backend is the internal working part of the web application. The backend of the application is responsible for things like calculations, business logic, database interactions, and performance.

Backend code is run on the server, as opposed to the client. Common backend technologies are Python, Ruby, Node.js and databases.

### 5.2.1 Introduction to technologies used

Node.js, Express.js and MongoDB were selected for the backend development of the project. Each of these technologies are briefly explained below.

#### Node.js

Node.js is an open source, cross-platform runtime environment for executing JavaScript code on the server. Node.js is useful for developing applications that require a persistent connection from the browser to the server and is often used for real-time applications such as chat, news feeds and web push notifications. [26]

```
const http = require("http");

const hostname = "127.0.0.1";
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader("Content-Type", "text/plain");
  res.end("Hello World\n");
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

FIGURE 16. HTTP server with plain Node.js [27]

As seen in the figure 16, a local server is created that runs in the port 3000 and returns a plain Hello World text.

## Express.js

Express.js is a web application framework for Node.js, released as free and open-source software under the MIT license. It is designed for building single-page, multi-page, and hybrid web applications and is the de facto standard server framework for node.js. [28]

```
const express = require("express");
const app = express();
const port = 3000;

app.get("/", (req, res) => res.send("Hello World!"));

app.listen(port, () => console.log(`Server running at ${port}!`));
```

FIGURE 17. Simple Express app

As seen in the figure 17, a simple Express application listening in the port 3000 is created that returns a plain Hello World text.

## MongoDB

MongoDB is a cross-platform and open source document-oriented database, a kind of NoSQL database. It has rapidly grown to become a popular database for web applications and is a perfect fit for Node.js applications. [29]

A record in MongoDB is a document, which is a data structure composed of field and value pairs. In MongoDB, data is stored in JSON-like documents. The values of fields may include other documents, arrays, and arrays of documents. [30]

MongoDB data model can be used to define the relationships, data arrays and other complex structures in the document as shown in the figure 18 below.

```
{
  _id: 111,
  title: '5G Field Test Web Applications',
  description: 'Full stack JavaScript applications for 5G field tests'
  stack: ['MongoDB', 'Express.js', 'Node.js', 'ReactJS'],
  contact: [
    {
      name: 'Suman Kharel',
      email: 'suman.kharel@kharelworld.com'
    },
    {
      name: 'John Smith',
      email: 'john.smith@smithworld.com'
    }
  ]
}
```

FIGURE 18. Sample MongoDB document

As seen in the figure 18, a JSON-like document stores the information of the 5G Field Test Application. A description, development stack and contact user list are stored in a sample MongoDB document.

## 5.2.2 Restful APIs

An API is the acronym of Application Programming Interface, which is a software intermediary that allows two applications to talk to each other. When the application is used, it connects to the internet and sends the data to a server. The server then retrieves that data, interprets it, performs the necessary actions and sends it back to the application. The application then interprets that data and present readable information to the user. All this process happens via an API.

REST is the acronym of Representational State Transfer, which is a web services API. REST APIs are based on Uniform Resource Identifies (URI) and the HTTP protocol and use JSON for the data format. Some of the advantages of RESTful APIs are fewer security requirements, browser client requirement, discoverability, data health and scalability. [31]

RESTful APIs were used to handle log files and corresponding user actions in the application.

```

// @route GET api/logs/files/details/id
// @desc get file details, sizes
router.get("/files/details/:id", (req, res) => {
  //send back error if db is not connected
  if (!gfs) {
    return res.status(503).json({ error: "No internet connection" });
  }
  gfs.files.findOne(
    { _id: mongoose.Types.ObjectId(req.params.id) },
    (err, file) => {
      if (!file) {
        return res
          .status(404)
          .json({ error: `File not found with id ${req.params.id}` });
      }

      if (err) {
        return res.status(503).json({
          error: `Error while loading details of ${
            file && file.filename ? file.filename : req.params.id
          }`
        });
      }

      return res.json(file);
    }
  );
});

```

FIGURE 19. API to get the file details

The figure 19 shows the RESTful API created to get the files details by its ID. The API returns error if the database is not connected or the file is not found. If the file was found, it returns the file in the JSON format.

As seen in the figure 19, when the GET request is made to the `api/logs/files/details/file_id` route, the API returns the details of the file. The figure 20 shows the response when the correct file identification was passed in the GET request.

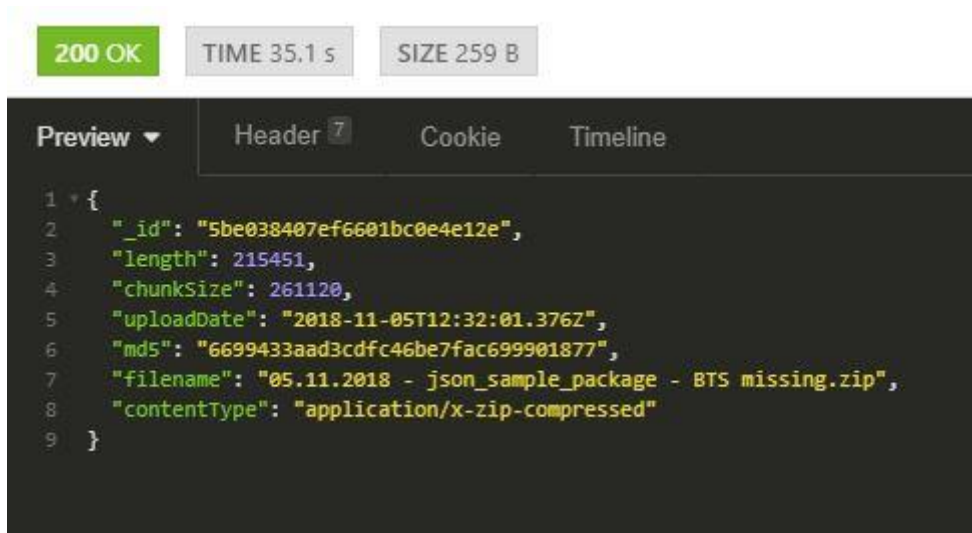


FIGURE 20. Response from the RESTful API in the insomnia tool

### 5.3 Frontend

The frontend is the visual representation of the application. It is what a user sees, feels and experiences while using the application. The following section explains the technologies that were used and parts of the dashboard's frontend.

#### 5.3.1 Introduction to technologies used

ReactJS was used as a frontend library with Redux for the state management of the application. Each of these technologies are briefly explained below.

##### ReactJS

ReactJS is a popular open source front-end JavaScript library developed by Facebook. React is widely popular among developer communities because of its simplicity and easy but effective developing process. React makes it easier to create interactive user interfaces. [32]

ReactJS uses the virtual Document Object Model (DOM) which is faster than the browser DOM. The browser DOM is the representation of an HTML elements in nodes and objects. DOM can be manipulated with the JavaScript to update the content of the application [33]. The virtual DOM is an in-memory representation of the real DOM elements generated by React components before any changes are made to the web pages.

```
import React, { Component } from "react";

class MyComponent extends Component {
  render() {
    return (
      <div>
        <h1>I am a simple react component</h1>
      </div>
    );
  }
}

export default MyComponent;
```

*FIGURE 21. A simple react component*

A typical React component is seen in the figure 21. It imports React and the component from a react module. A class `MyComponent` is created which renders a div with a heading as seen in the figure 22.

I am a simple react component

*FIGURE 22. The Rendered react component*

The heart of every React component is its state, an object that determines how the component renders and behaves. React is built in a way for components to internally manage its state without any need for an external library or tool. It does well for applications with few components but as the application grows bigger, managing states shared across components becomes a chore.

In an app where data is shared among components, it might be confusing to know where a state should live. Ideally, the data in a component should live in just one component. So, sharing data among sibling components becomes difficult. Therefore, the bigger applications need a state management tool such as Redux to maintain these states. The following section explains the Redux and its concepts.

## Redux

Redux is an open-source JavaScript library for managing application state. Redux not only aids in the state management but makes it straightforward to implement advanced features such as infinite undo/redo and live editing time travel. [34]

Redux is founded on three principles

1. Single source of truth.

Redux manages the state through a single JavaScript object, known as the data store, where all the application state is located. The centralized state in a single object makes it easier to reason about application data when reading code, debugging and testing it.

2. State is read-only

State can only be updated by emitting an action. An action is an object describing what happened. This ensures that neither the views nor the network callbacks will ever write directly to the state, and instead express the intent to transform the state.

3. Reducers create the next state, given the current state and an action.

Reducers are just pure functions that take the previous state and an action and return the next state. Because they are pure functions, reducers are easy to read, test and debug. [35]

Understanding Redux concepts and data flow might be confusing, especially if a developer starts using it without experiences with the React internal state. The figure 23 shows the common data flow used in the Redux.



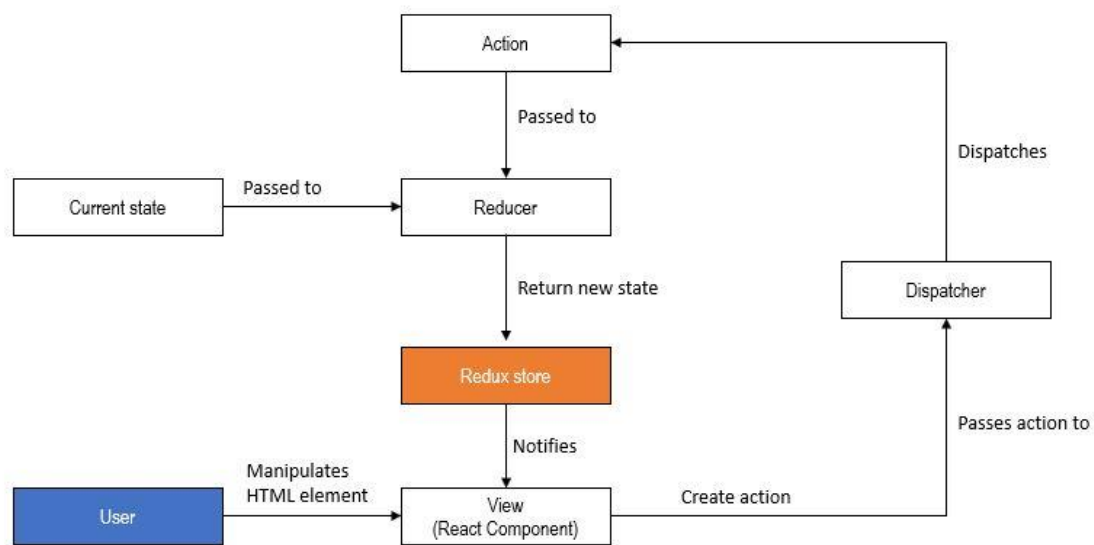


FIGURE 23. The Redux data flow [35]

As seen in the figure 23, when the user manipulates an HTML element inside the application, the action is passed to the dispatcher function. The dispatched action alters the initial or existing state inside the reducer function, which is passed as a new state to the Redux store. Redux notifies the changes to the view component and the component is updated.

The 5G test dashboard frontend can be divided into five parts. The Main menu, Map view, Calendar view, Selected files view and Drone 3D view. The following section describes each of these parts with screenshots for the better illustration.

### 5.3.2 Main Menu

The main menu is the landing page of the dashboard. It includes options to upload files and lists all the uploaded files. As seen in the figure 24, the main menu is composed of header, filter, upload logs form, logs list, breadcrumbs components.

The screenshot shows the '5G TEST DASHBOARD > MENU' interface. At the top, there is a navigation bar with 'Menu', 'Drive', and 'Drone' icons, and a user profile for 'Suman Kharel' with a 'Log out' button and the 'NOKIA' logo. Below this, the 'Select Log' section features a sidebar on the left with filters for 'Sitename', 'Radio Software', and 'Brand Model'. The main area contains a file upload section with a 'Choose File' button, 'No file chosen' text, and an 'Upload' button. To the right, there is a 'Show Logs' dropdown set to '20' and a search box. The central part of the dashboard is a table with the following columns: 'Upload Date', 'Test Date', 'SwPackage', 'BrandModel', 'Test Name', 'File Name', 'File Size', and 'Delete'. The table contains 10 rows of log entries, each with a checkbox in the 'Upload Date' column and a trash icon in the 'Delete' column.

Upload Date	Test Date	SwPackage	BrandModel	Test Name	File Name	File Size	Delete
05.11.2018, 14:32:01	28.06.2018, 16:46:22	swPackage → 3.0.178360	brandModel → Intel MTP	Some test name that is sooo long, is it long enoug...	05.11.2018 - json_sample_package - BTS missing.zip	210.4 KB	🗑️
<input checked="" type="checkbox"/>	20.08.2018, 13:51:46	swPackage → 3.0.178360	brandModel → Intel MTP	Some test name	20.08.2018 - json_sample_package.zip	280.09 KB	🗑️
	16.08.2018, 13:07:50	swPackage → 3.0.178360	brandModel → Intel MTP	Some test name	16.08.2018 - json_sample_package.zip	280.09 KB	🗑️
<input checked="" type="checkbox"/>	08.08.2018, 13:48:53	swPackage → 3.0.178360	brandModel → Intel MTP	Some test name	08.08.2018 - json_sample_package.zip	280.09 KB	🗑️
<input checked="" type="checkbox"/>	26.07.2018, 11:11:45	swPackage → 3.0.178360	brandModel → Intel V2	Some test name	26.07.2018 - short.zip	280.45 KB	🗑️
	24.07.2018, 16:22:03	swPackage → 3.0.178360	brandModel → Intel MTP	Some test name	24.07.2018 - json_sample_package.zip	280.09 KB	🗑️
	24.07.2018, 16:21:21	swPackage → 3.0.178360	brandModel → Intel MTP	Some test name	24.07.2018 - json_sample_package.zip	280.09 KB	🗑️
	24.07.2018, 15:51:37	swPackage → 3.0.178360	brandModel → Intel MTP	Some test name	24.07.2018 - json_sample_package.zip	280.09 KB	🗑️
	24.07.2018, 15:45:34	swPackage → 3.0.178360	brandModel → Intel V2	Some test name	24.07.2018 - json_sample_package - Copy.zip	280.09 KB	🗑️

FIGURE 24. The 5G test dashboard - main menu

### Header Component

The header component was designed to meet Nokia’s UI look and feel guidelines. It displays the navigation links, name of company and username.

### Filter Component

One of the main requirements of the project was to develop an extensive filter so that the user can find the intended log file effortlessly. The sidebar with the site name, radio software and the brand model dropdown options enable a user to filter the logs. The search box on the top right in the figure 24 can be used to filter logs by text inputs. As seen in the figure 25, the logs are filtered based on the site name, software model and upload date.

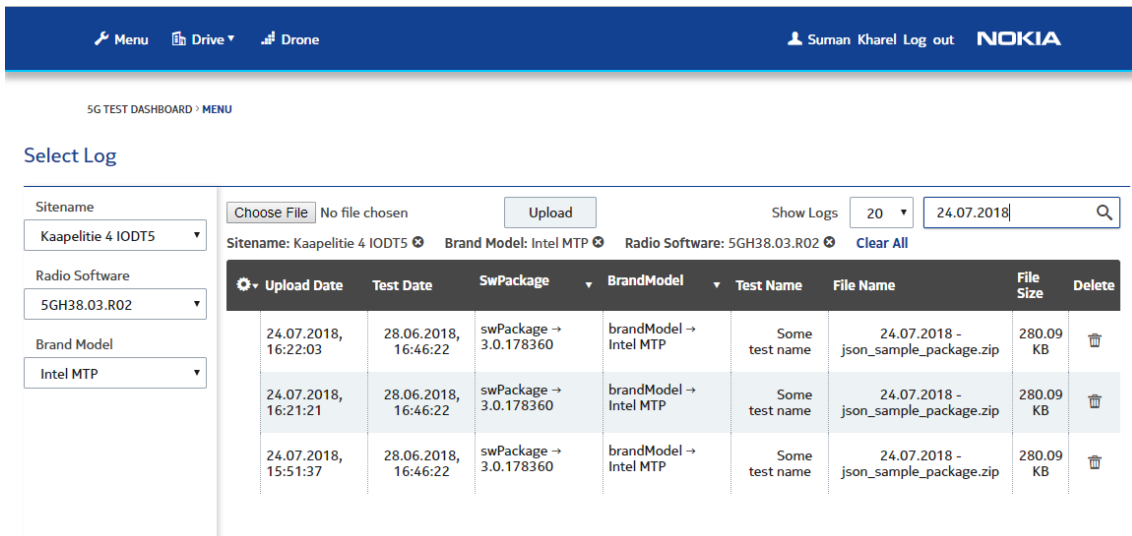


FIGURE 25. The list of the filtered log files

### Upload Logs Form Component

The Upload Logs component enables a user to upload the log files obtained from the field test to the dashboard's database. The log file is validated before uploading it to the database. The validation function checks that the file structure is consistent, and that the required files exists. If the required files are missing, an error message is displayed on the dashboard. If some optional files are missing, a dialogue box with an option to continue is displayed. The figure 26 shows the dialogue box when the optional files are missing.

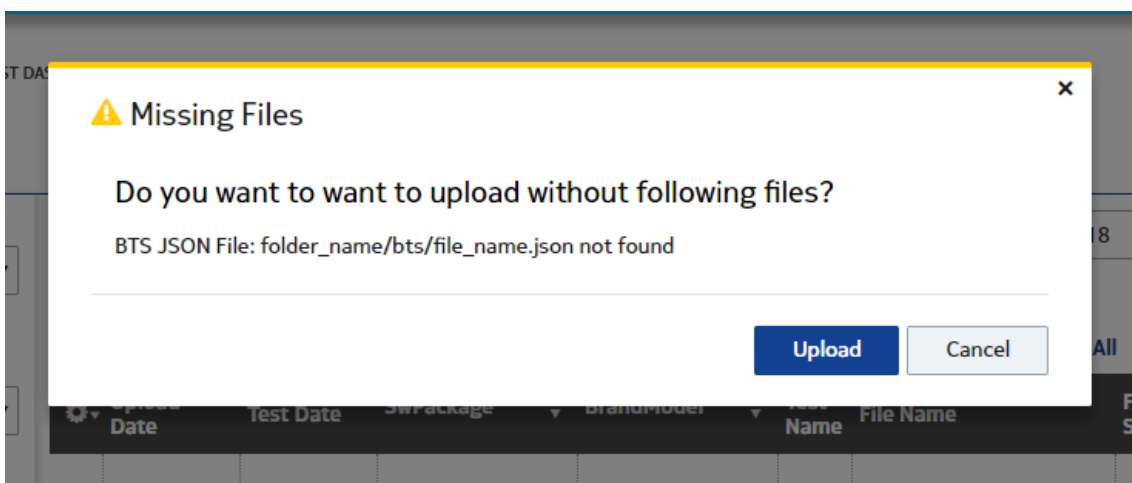


FIGURE 26. The warning dialogue for a missing optional file

### Logs List Component

The Logs List component, as implied by the name of component, lists all the uploaded log files. Listed logs show the upload date, test date, software package, brand model, test name, file name,

file size and delete option. When the delete icon is clicked, a dialogue box is displayed as seen the figure 27. A user can confirm or cancel the deletion of the log file.

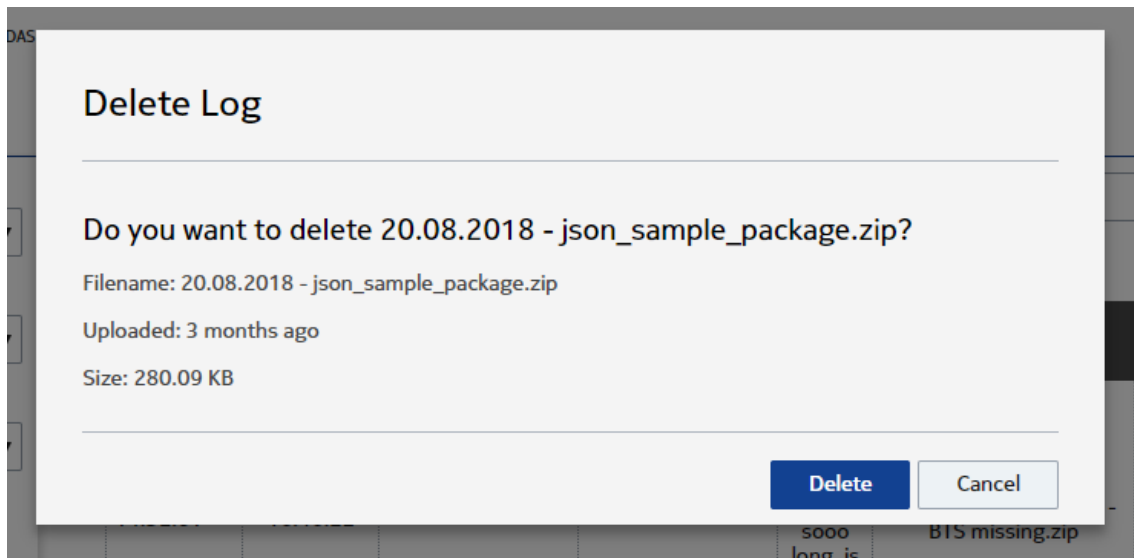


FIGURE 27. The Delete Log confirmation dialogue

A user can select a log file by clicking on the list to visualize data inside it. As seen in the figure 28, a dialogue is displayed on the dashboard to confirm the visualization.

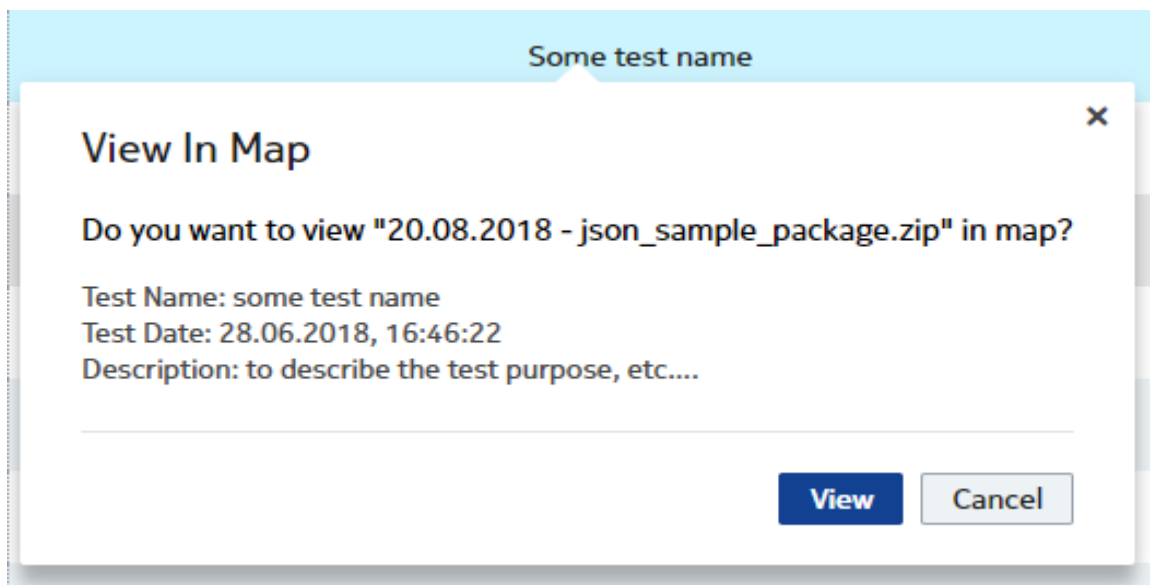


FIGURE 28. The View in Map tooltip

### 5.3.3 Map View

A Map View is the page displayed after the confirmation of data visualization in the figure 28. It includes an option to visualize a selected log file on the map. As seen in the figure 29, a map view is composed of a map component and form components.

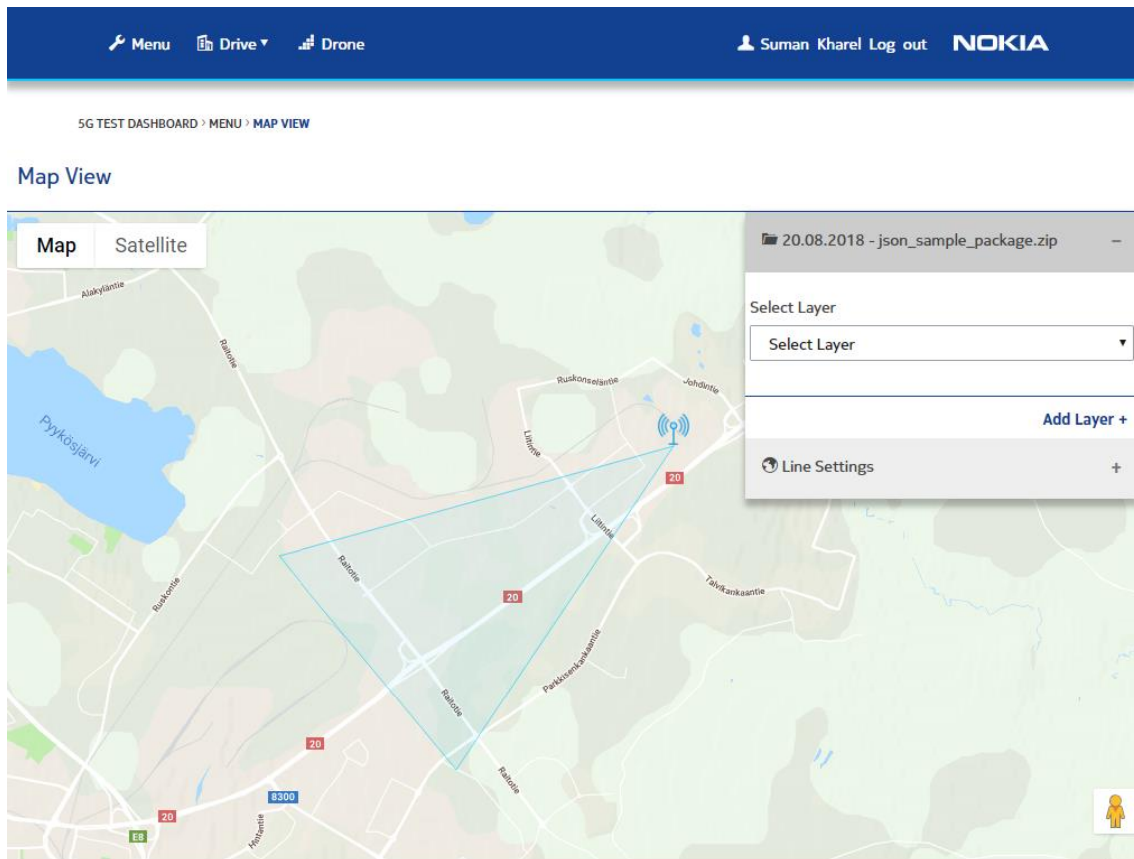


FIGURE 29. The 5G test dashboard - map view

#### Map Component

A Map Component is used to visualize the time-based log file obtained from the drive test. As seen in the figure 29, the initially loaded map component includes an icon with the lines drawn through it. An icon indicates the location of 5G tower and lines indicate the coverage area and the beam direction of tower.

#### Form Component

A form component displays the selected file name, select layer options, and line settings component. It enables a user to select the layers to visualize and to update visualized data.

As seen in the figure 30, when the ABIK 1CC BLER layer was selected, a polyline was drawn over the map to display data obtained from the test drive. Good and excellent values, which can be updated manually, were automatically generated by the application based on the data from the selected layer. Any data below the good value was visualized with a red line. Any data between the good and the excellent value was visualized with a yellow line and data above the excellent value was visualized with a green line.

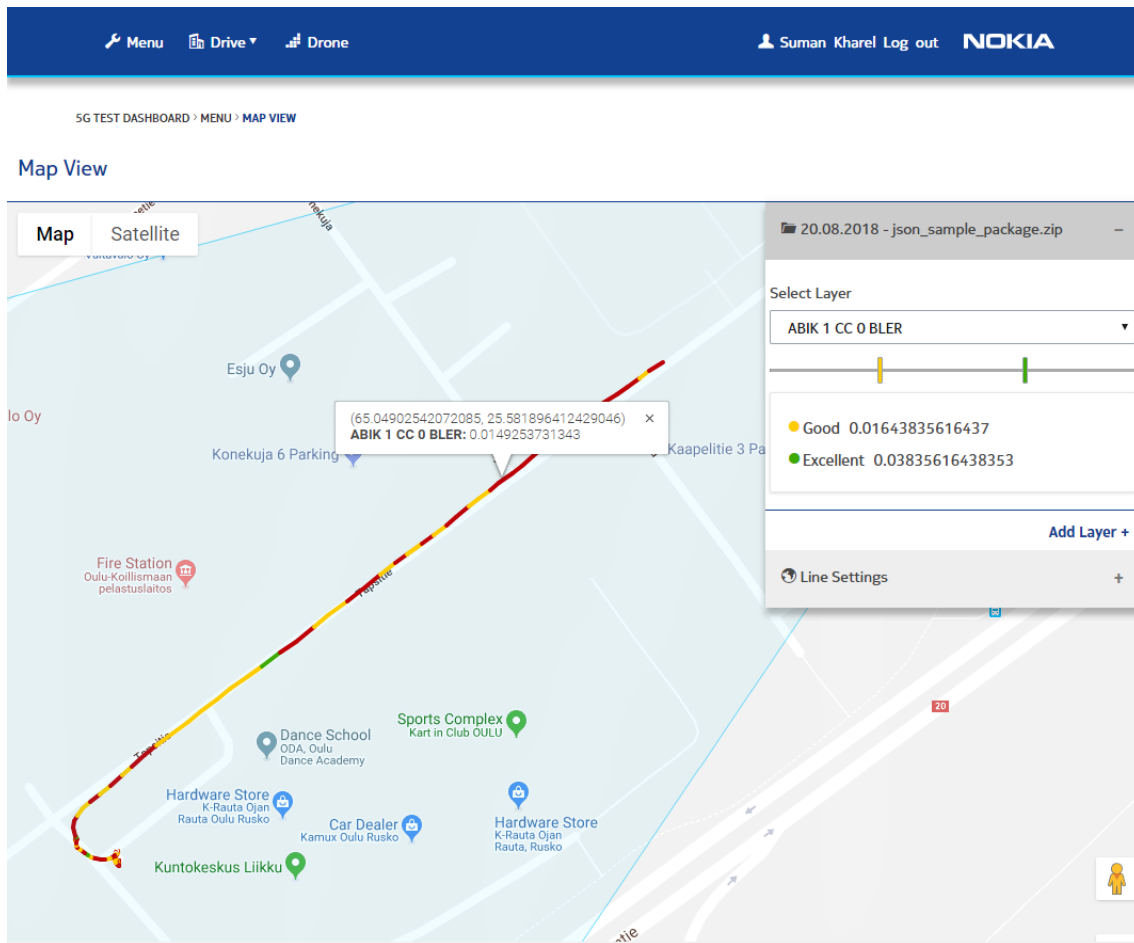


FIGURE 30. The data visualization on the map

Data can also be visualized on replay mode to mimic the actual drive from Line Settings component. In addition, Line Settings components include the features to update the visualized line's thickness and opacity. The figure 31 shows the snapshot of data visualization on replay mode with the line thickness increased to sixteen.

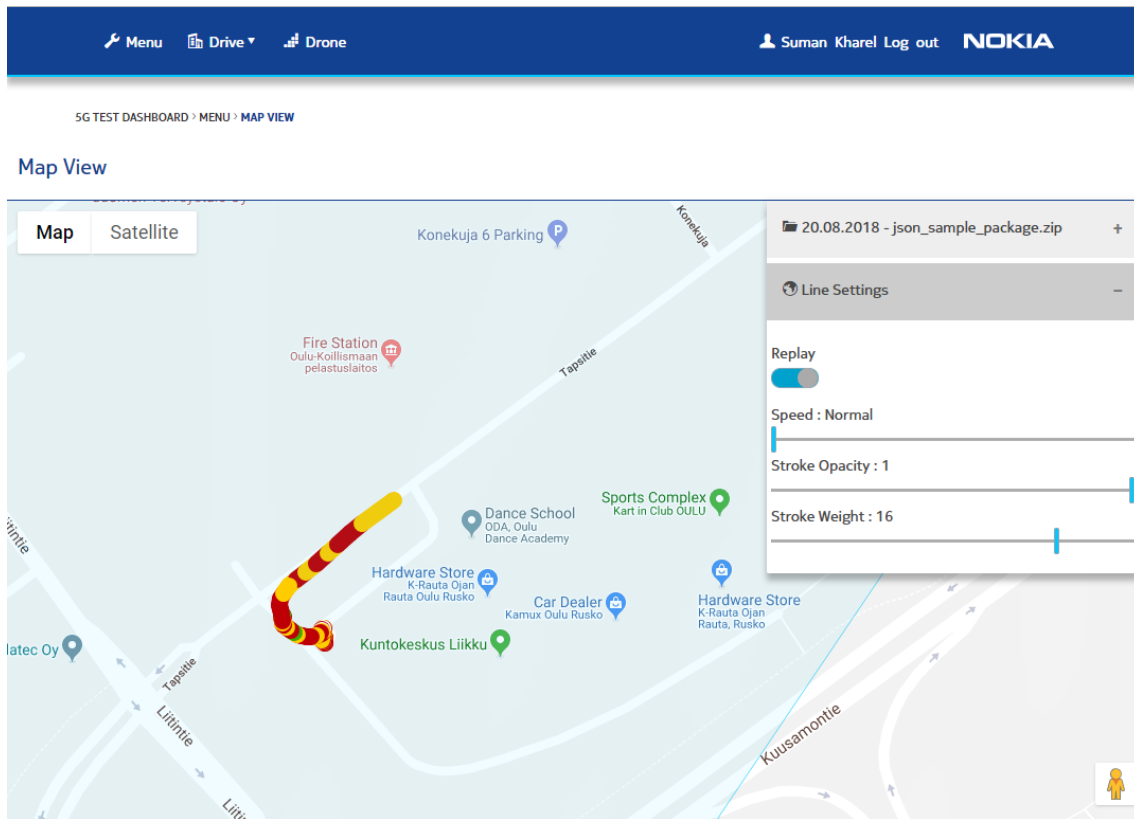


FIGURE 31. Data visualization on replay mode

A map view component visualizes a selected log file. As mentioned in the application requirements in the section 5.1.2, a user should be able to compare multiple log files. The following section describes the implementation of this feature.

### 5.3.4 Calendar View

A Calendar View is one the pages to visualize multiple log files. The other page is explained in the section 5.3.5. A Calendar View page includes the option to select the total number of the latest log files to visualize. Once the number is selected, a user can select the layer.

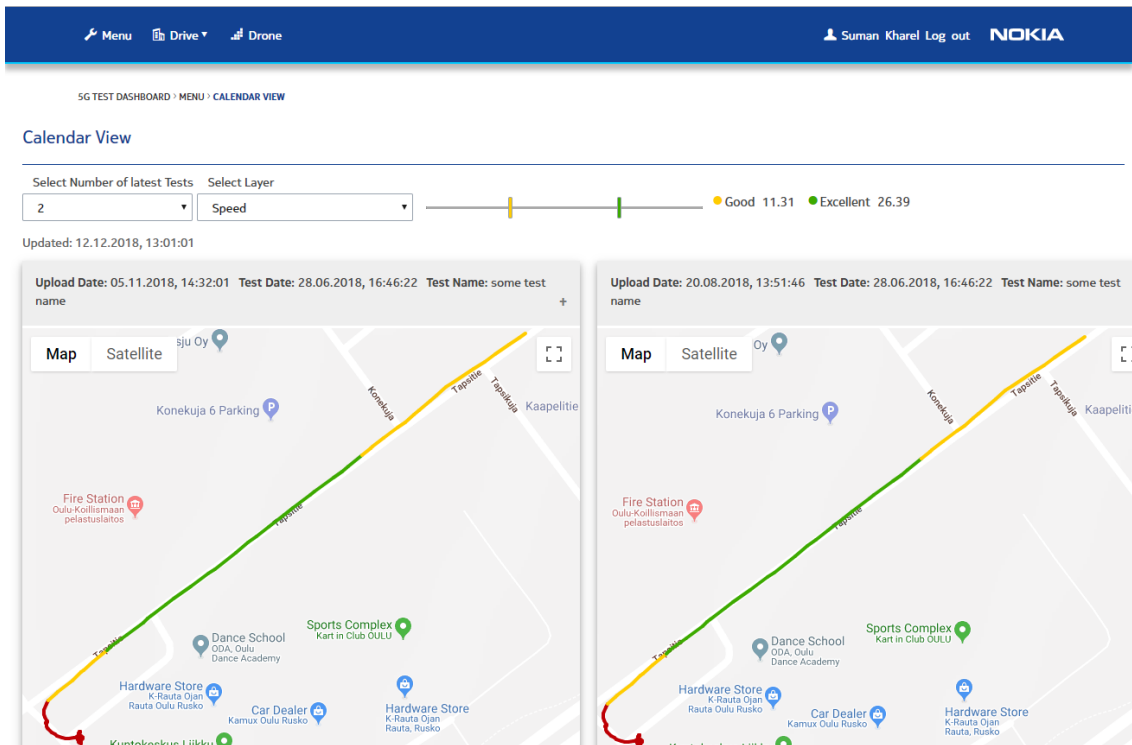


FIGURE 32. Visualization of multiple log files

As seen in the figure 32, the latest two log files are visualized. The layer was selected, and the visualized layer was comparable. While this feature enabled a user to compare multiple log files, a user was not able to select the log files from the list to visualize. The following section describes the implementation of this feature.

### 5.3.5 Selected Files View

A Selected Files View is the page displayed when the user selects multiple log files from the main menu to visualize. It includes the option to select the layer and update the good and excellent points of the selected layer.



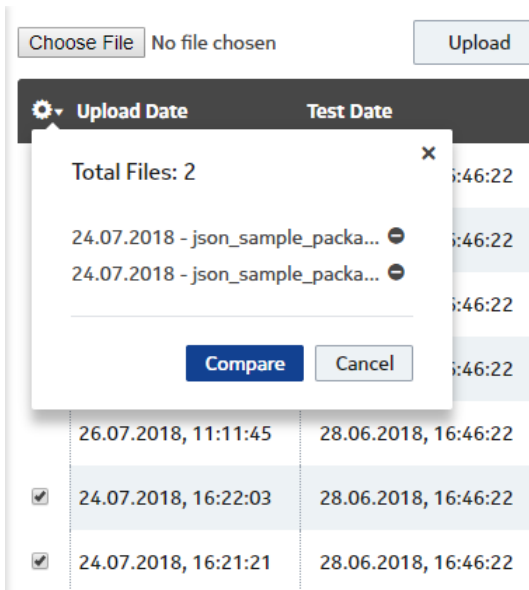


FIGURE 33. The selected files tooltip

As seen in the figure 33, a user can select the log files from the list to compare. the selected log files were displayed on the dashboard. The layer ABIK 1CC 0 BLER was selected and compared on the map as seen in the figure 34.

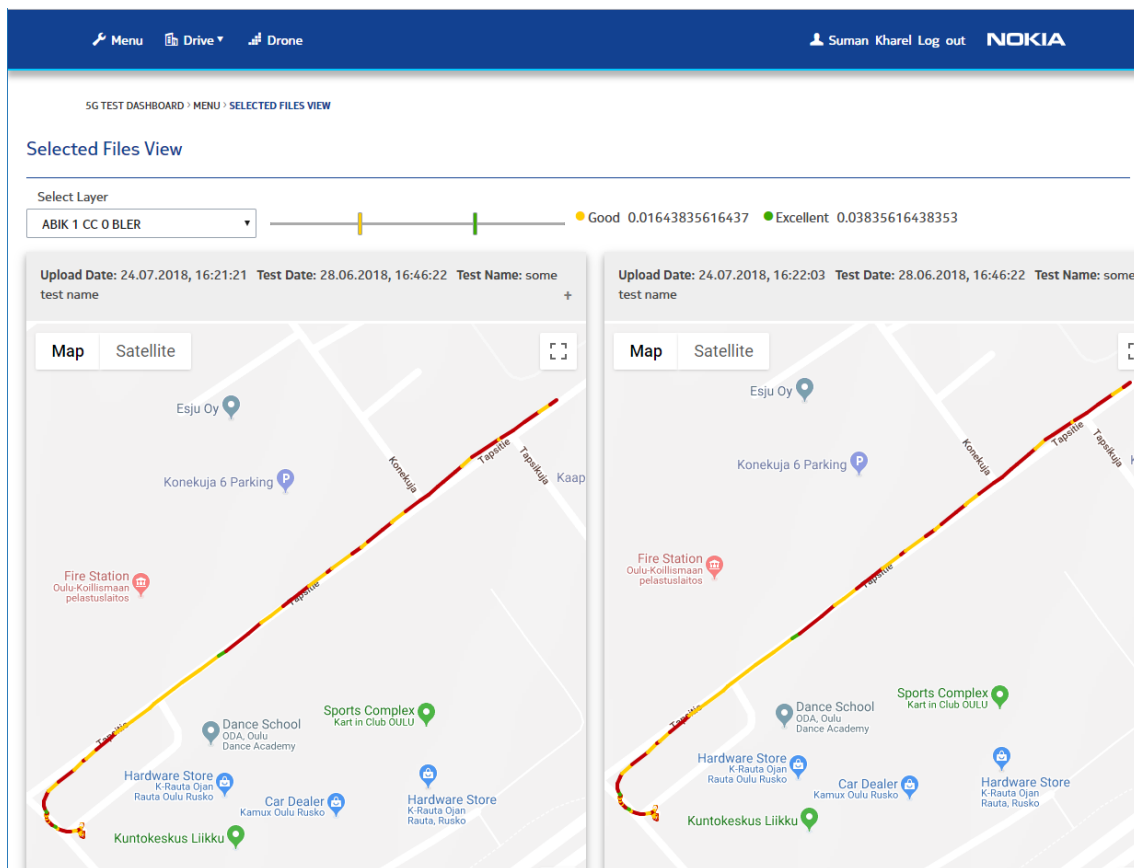


FIGURE 34. The visualization of selected log files

The visualization on the map was developed for the log files obtained from the field test by car. The team also has multiple drones to measure the network in the altitude. The following section explains the 3D view of log files obtained from the field test by drones.

### 5.3.6 Drone 3D View

A Drone 3D View page visualizes logs obtained from the field test by drone in 3D. As seen in the figure 35, a log file is visualized in 3D with X, Y and Z points. The line can be hovered to view the data at that point.

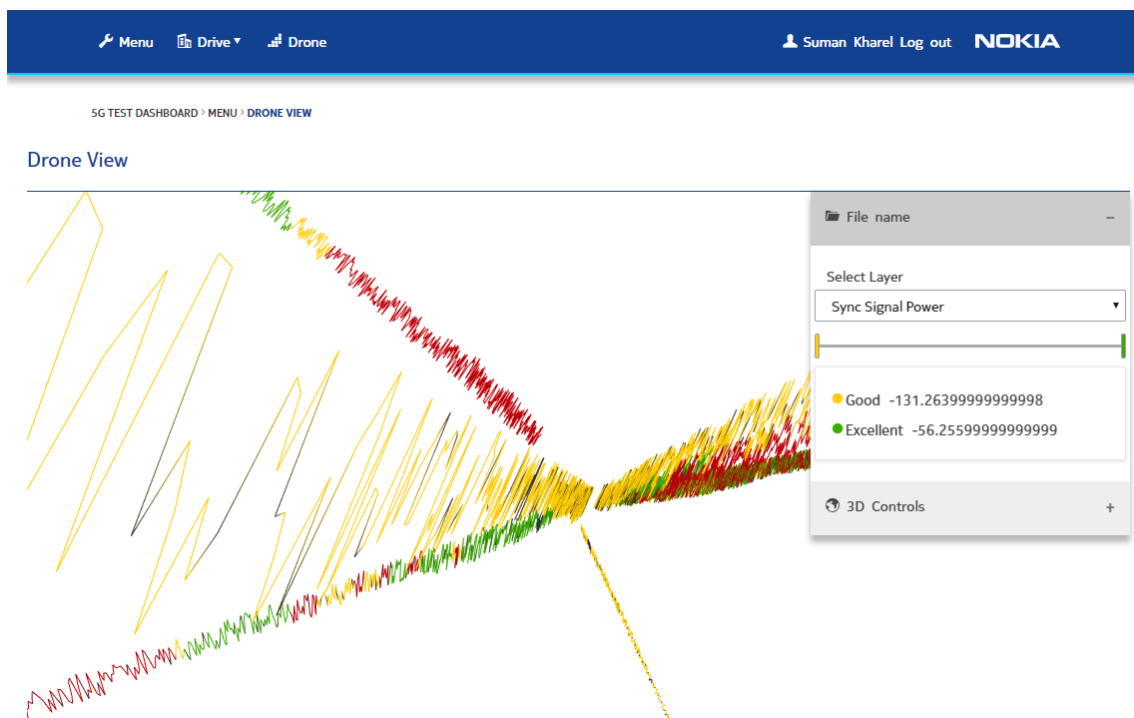


FIGURE 35. The 3D data visualization

The 3D view is in its initial phase. The view will be updated according to the future requirements.

## 6 CONCLUSION

A complete application was developed and delivered to Nokia's 5G team on time. I learned how to develop full stack JavaScript applications from scratch.

Several meetings were held with Nokia's 5G team during the development period to show the progress and discuss possible next steps. The application was developed according to the team's suggestions and initial requirements.

A 5G test dashboard is deployed to Nokia's internal network. The application is used by the team to upload log files obtained from the field test. An extensive filter system was implemented to enable a user to find the updated files effortlessly. Log files can be selected from the list to visualize. A 3D view was developed to visualize data by drones in three dimensions.

## REFERENCES

1. Wikipedia 2018. Nokia. Date of retrieval 09.10.2018  
<https://en.wikipedia.org/wiki/Nokia>
2. Nokia 2018. Our history. Date of retrieval 04.12.2018  
<https://www.nokia.com/about-us/who-we-are/our-history/>
3. Carritech telecommunications 2017. The evolution of mobile communications, from 1G to 5G. Date of retrieval 09.10.2018  
<http://www.carritech.com/news/evolution-mobile-communication-1g-5g/>
4. Nokia 2018. Be prepared for 5G. Date of retrieval 04.12.2018  
<https://networks.nokia.com/5g/get-ready>
5. The Tech Terms Computer Dictionary 2014. Web application. Date of retrieval 17.10.2018  
[https://techterms.com/definition/web\\_application](https://techterms.com/definition/web_application)
6. Microsoft 2018. Common client-side web technologies. Date of retrieval 12.10.2018  
<https://docs.microsoft.com/en-us/dotnet/standard/modern-web-apps-azure-architecture/common-client-side-web-technologies>
7. MDN web docs 2018. What is JavaScript? Date of retrieval 11.10.2018  
<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction>
8. Medium 2016. Single-page applications vs Multi-page applications. Date of retrieval 11.10.2018  
<https://medium.com/@NeotericEU/single-page-application-vs-multiple-page-application-2591588efe58>
9. Codementor community 2017. Modern web development approaches. Date of retrieval 11.10.2018

<https://www.codementor.io/savinda/modern-web-development-approaches-fbkb7xnm>

10. Wikipedia 2018. Progressive web applications. Date of retrieval 11.10.2018  
[https://en.wikipedia.org/wiki/Progressive\\_web\\_applications](https://en.wikipedia.org/wiki/Progressive_web_applications)
11. Google Developers. The web app manifest. Date of retrieval 11.12.2018  
<https://developers.google.com/web/fundamentals/web-app-manifest/>
12. Visual Studio Code. Code editing, redefined. Date of retrieval 05.12.2018  
<https://code.visualstudio.com/>
13. Atom. A hackable text editor for the 21<sup>st</sup> century. Date of retrieval 05.12.2018  
<https://atom.io/>
14. Brackets. A modern, open source editor that understands web design. Date of retrieval 05.12.2018  
<http://brackets.io/>
15. Sublime. A sophisticated text editor for code, markup and prose. Date of retrieval 05.12.2018  
<https://www.sublimetext.com/>
16. Git. Free and open source distributed version control system. Date of retrieval 05.12.2018  
<https://git-scm.com/>
17. CVS. Date of retrieval 05.12.2018  
<https://www.nongnu.org/cvs/>
18. Mercurial. Free and distributed source control management tool. Date of retrieval 05.12.2018  
<https://www.mercurial-scm.org/>
19. Bazaar. Date of retrieval 05.12.2018  
<http://bazaar.canonical.com/en/>

20. GitLab 2018. Nokia 5G test dashboard. Date of retrieval 15.10.2018  
<https://gitlab1.ext.net.nokia.com/kharel/5G-test-dashboard/commits/master>
21. Insomnia. Insomnia Documentation. Date of retrieval 05.12.2018  
<https://support.insomnia.rest/>
22. REST Assured. Date of retrieval 05.12.2018  
<http://rest-assured.io/>
23. SoapUI. Build better, Test smarter. Date of retrieval 05.12.2018  
<https://www.soapui.org/>
24. JMeter. Date of retrieval 05.12.2018  
<https://jmeter.apache.org/>
25. Postman. API development environment. Date of retrieval 05.12.2018  
<https://www.getpostman.com/>
26. Wikipedia 2018. Node.js. Date of retrieval 15.10.2018  
<https://en.wikipedia.org/wiki/Node.js>
27. Node.js 2018. About Node.js. Date of retrieval 15.10.2018  
<https://nodejs.org/en/about/>
28. Express. Node.js web application framework. Date of retrieval 15.10.2018  
<https://expressjs.com/>
29. MongoDB. Open source document database. Date of retrieval 17.10.2018  
<https://www.mongodb.com/>
30. Panoply 2017. MongoDB vs MySQL. Date of retrieval 17.10.2018  
<https://blog.panoply.io/mongodb-and-mysql>

31. Stack Overflow. What exactly is RESTful programming? Date of retrieval 17.12.2018  
<https://stackoverflow.com/questions/671118/what-exactly-is-restful-programming>
  
32. ReactJS. A JavaScript library for building web interfaces. Date of retrieval 17.12.2018  
<https://reactjs.org/>
  
33. MDN. Introduction to the DOM. Date of retrieval 12.12.2018  
[https://developer.mozilla.org/en-US/docs/Web/API/Document\\_Object\\_Model/Introduction](https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction)
  
34. Redux. A predictable state container for JS apps. Date of retrieval 17.10.2018  
<https://redux.js.org/>
  
35. IBM 2016. Introducing Redux. Date of retrieval 17.10.2018  
<https://www.ibm.com/developerworks/library/wa-manage-state-with-redux-p1-david-geary/index.html>