

INTERNSHIP REPORT
ON
EDITABLE XML DATA ON WEB
At



DEFENCE RESEARCH AND DEVELOPMENT LABORATORY
P.O KANCHANBAGH, HYDERABAD
IN PARTIAL FULFILLMENT OF THE DEGREE IN BACHELORS OF
TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING
GANDHI INSTITUTE OF TECHNOLOGY AND MANAGEMENT

Submitted By
Akash Varma Saripella (HU21CSEN0100672)

UNDER GUIDANCE
OF
KANAN BALA SAHOO

Acknowledgment

We would like to express our sincere gratitude to everyone who has contributed to the successful completion of this internship project.

First and foremost, we would like to thank Smt **Kanan Bala Sahoo, Sc 'F'**, DWST/DRDL for her invaluable guidance, support, and encouragement throughout the project. Her insights and expertise were crucial to the development and completion of this work.

We are also grateful to the entire team at DRDO for providing us with the opportunity to undertake this internship. Their resources and support were instrumental in achieving the project goals.

A special thank goes to our colleagues and friends, whose constant support and constructive feedback have greatly enhanced the quality of this project. Their collaboration made the process of creating an editable webpage that displays a table made of XML both enriching and enjoyable.

Thank you all.

Abstract

This project focuses on the development of an editable webpage designed to manage and modify the contents of a table created using XML. The project was undertaken as part of an internship at DRDO. The initial phase involved a thorough exploration of various command and control systems, aiming to understand their functionalities and potential applications. Following this, an investigation into various XML tools compatible with the Linux environment was conducted to identify the most suitable ones for our project needs.

The primary objective of the project was to create a dynamic and user-friendly webpage that allows admin to edit table contents efficiently. The webpage was developed using a combination of XML for table creation and other web technologies such as JavaScript and Node.js for functionality and interactivity. The end result is an intuitive interface that simplifies the process of editing XML-based tables, enhancing user experience and productivity.

This project not only provided valuable insights into command and control systems and XML tools but also equipped us with practical experience in web development and the integration of different technologies. The collaborative effort and guidance from our supervisors played a crucial role in the successful completion of this project.

Table of Contents:

Contents	Page Numbers
Acknowledgement	2
Introduction	5
Internship Objectives	5
Work and Projects	
<input type="checkbox"/> Command and Control Systems	6
<input type="checkbox"/> Prototype 1	11
<input type="checkbox"/> Prototype 2	15
Learning and Development	16
Challenges and Solutions	16
Environment Variable Setting	20
Conclusion	23

1. Introduction

Organization Overview

The Defence Research and Development Laboratory (DRDL) is a prestigious research facility under the auspices of the Defence Research and Development Organisation (DRDO), situated in Hyderabad, India, DRDL specializes in the research, design, and development of advanced missile systems and related technologies.

Internship Duration

The internship at DRDL spanned a period of two months, commencing from May 2024 to July 2024. During this time, I had the opportunity to immerse myself in various projects and gain invaluable insights into the field of missile technology and research.

Internship Objectives

1. Discover and Understand Various Command and Control Systems: The internship aimed to explore a range of command and control systems, including radar, satellite communications, and unmanned aerial systems. The focus was on understanding their technical aspects and integration.

2. Explore and Test Various XML Tools Compatible with the Linux Environment: This objective involved evaluating XML tools for Linux, focusing on parsers, editors, and validators. The aim was to find tools that efficiently handle XML data and meet various application needs.

3. Create a Static XML Editor for One Specific File: The task was to develop a specialized static XML editor for a specific file format, featuring functionalities like syntax highlighting and schema validation. The goal was to provide a user-friendly tool for managing this format.

4. Create a Dynamic XML Editor for Any File: The final objective was to create a dynamic XML editor capable of handling various XML files. This tool needed to support features like tag completion and error correction, offering flexibility and ease of use for diverse applications.

Research and Prototypes

Command and Control Systems

Radar Systems:

We explored various radar systems, focusing on their functionalities, operational frequencies, and applications in both civilian and defense sectors. This includes ground-based, airborne, and maritime radar systems used for surveillance, tracking, and targeting. The study also covered advancements in radar technology, such as phased array systems, and their role in modern warfare and air traffic control.

Satellite Communication Systems:

We studied satellite communication systems, examining different communication protocols, frequency bands (such as L, S, C, X, Ku, and Ka bands), and data transmission methods. The exploration included the role of geostationary, medium, and low Earth orbit satellites in global communication networks. Additionally, the study delved into the infrastructure required for ground stations and the integration of satellite systems with terrestrial networks.

Unmanned Aerial Systems (UAS):

Our analysis of Unmanned Aerial Systems covered the various control mechanisms, including remote piloting and autonomous operations. We explored the operational frameworks, including mission planning, navigation, payload integration, and data collection. The study also addressed regulatory and safety considerations, as well as the diverse applications of UAS in military reconnaissance, surveillance, and civilian sectors such as agriculture and disaster management.

Electronic Warfare (EW):

We investigated the techniques and technologies used in electronic warfare, focusing on the strategies for electronic attack, protection, and support. The study included an analysis of electronic jamming, deception,

and countermeasures, as well as the use of EW systems to disrupt enemy radar, communication, and navigation systems. We also looked at the role of signals intelligence (SIGINT) and electronic intelligence (ELINT) in EW operations.

Cyber Defense Systems:

Our investigation into cyber defense systems involved a comprehensive study of cybersecurity measures, including firewalls, intrusion detection and prevention systems, and encryption protocols. We explored the defense protocols for protecting critical infrastructure and sensitive information from cyber threats. The study also addressed the challenges in cyber defense, such as threat intelligence, incident response, and the evolving landscape of cyber attacks.

Geographic Information Systems (GIS):

We examined the role of Geographic Information Systems in managing and analyzing spatial data. The study included the use of GIS for mapping, geospatial analysis, and decision-making in various sectors, including urban planning, environmental monitoring, and defense. We explored the integration of GIS with other technologies, such as remote sensing and GPS, to enhance data accuracy and usability.

Integrated Air and Missile Defense Systems (IAMD):

The study of Integrated Air and Missile Defense Systems focused on the coordination of multiple systems to detect, track, and intercept aerial threats. We explored the integration of radar, missile systems, and command and control centers to form a cohesive defense network. The study also included an analysis of the challenges in integrating different systems and technologies to provide a unified response to threats.

Battle Management Systems (BMS):

We explored the command and control processes within Battle Management Systems, focusing on their role in situational awareness, decision-making, and coordination of military forces. The study included

an analysis of the software and hardware components of BMS, as well as the integration of sensors, communication networks, and data processing systems. We also looked at the challenges in maintaining real-time data accuracy and ensuring secure communication in dynamic battle environments.

.

XML Development Tools and Approaches

1. Using jQuery and VS Code

- ❑ **Interface Creation:** A simple interface was designed to display XML data in an HTML table.
- ❑ **JavaScript Functionality:** The main function involves translating and displaying XML data.

2. Using Eclipse for XML Development

- ❑ **Advantages:**
 1. **IDE Features:** Offers a robust code editor with syntax highlighting, code completion, and error detection.
 2. **Plugins and Extensibility:** Supports specialized XML plugins and integrates with other development tools.
 3. **Schema and Validation Support:** Provides real-time validation against DTDs/XSDs with robust XML Schema support.
 4. **GUI Options:** Includes a tree view for XML structure and allows switching between design and source views.
 5. **Debugging and Testing:** Integrated debugging tools and testing frameworks.
 6. **Version Control Integration:** Seamlessly integrates with version control systems like Git and SVN.
 7. **Cross-Platform Availability:** Available on Windows, macOS, and Linux.

3. Using Atom for XML Development

- ❑ **Advantages:**
 - Highly customizable and supports various plugins and packages for XML development.
 - Popular packages:
 - `linter-xmlint`: Provides on-the-fly XML validation.
 - `xml-formatter`: Helps format XML files for better readability.
- ❑ **Validation Process:**
 - Create and save an XML file in Atom with a `.xml` extension (e.g., `library.xml`).
 - Use the built-in terminal in Atom to navigate to the file's directory.
 - Run `xmlint` to validate the XML file for syntax errors and validation issues.

Advantages of Using Node.js as a Backend for XML Tools in a Linux Environment

1. **Asynchronous and Event-Driven:** Node.js is non-blocking, making it efficient in handling multiple concurrent requests, which is especially useful for real-time applications.
2. **Scalability:** Its event-driven architecture and asynchronous I/O make Node.js well-suited for scaling applications.
3. **Cross-Platform Compatibility:** Node.js can run on various operating systems, including Linux, making it versatile for development and deployment.
4. **NPM Ecosystem:** A vast library of packages available through npm (Node Package Manager) allows developers to easily integrate various functionalities, such as XML parsing and validation.
5. **Performance:** Node.js's V8 engine ensures fast execution of JavaScript code, which can be beneficial for tasks like parsing and processing large XML files.
6. **Unified Development:** Developers can use JavaScript both on the frontend and backend, streamlining the development process and reducing the learning curve.
7. **Active Community and Support:** Node.js has a large and active community, offering extensive resources, modules, and support.

These features make Node.js a strong choice for developing XML tools in a Linux environment, providing efficiency, scalability, and ease of use.

Prototype 1:

Static X Editor Web Application

For our second task, we developed an XML editor web application using Node.js and various web technologies. This application allows users to manage a list of books stored in an XML format, providing functionalities for viewing, editing, and deleting book entries.

Project Overview

Project Directory Structure:

```
xml-tool/  
├── public/  
│   ├── data.xml  
│   ├── index.html  
│   ├── script.js  
│   └── styles.css  
├── server.js  
└── package.json
```

Components of the Project:

1. Data.xml

The `data.xml` file contains the initial list of books in XML format, including attributes such as title, author, and year.

2. Index.html

The HTML file provides the user interface for the XML editor, displaying a table of books with options to edit details and delete entries.

3. Script.js

The JavaScript file handles dynamic generation of the table content, editing of XML data, and saving changes back to `data.xml`.

4. Styles.css

The CSS file provides styling for the web application, ensuring a user-friendly interface.

5. Server.js

The Node.js server file manages requests and responses, allowing users to save edited XML data.

Functionality and Features:

- **Display and Edit:** Users can view and edit book details directly in the web interface.
- **Delete:** Option to remove books from the list.
- **Save Changes:** Button to save edited data back to `data.xml` on the server.
- **Dynamic Updates:** Table updates dynamically reflect changes made by the user.

Disadvantages of Static XML Editor Web Application

1. **Limited Functionality:**
 - Static web applications are often limited to displaying data and basic operations. They cannot dynamically update content without a full page refresh.
2. **Poor Scalability:**
 - Static applications are less flexible in handling complex, large-scale data manipulations and dynamic content updates.
3. **Lack of Real-Time Updates:**
 - Static applications don't support real-time updates, making it challenging to reflect changes instantly across multiple clients or sessions.
4. **Manual Data Handling:**
 - Editing XML data requires manual changes to the source files, which can be error-prone and inefficient, especially for large datasets.
5. **No User-Specific Customizations:**
 - Static applications typically lack user-specific data handling, such as personalized settings or saved states, which can be a drawback for complex applications.
6. **Limited Interaction and User Experience:**
 - User interactions are usually limited to basic inputs and do not support advanced functionalities like drag-and-drop, real-time collaboration, or instant feedback.
7. **Difficulty in Maintenance and Updates:**

- Making updates or adding new features often requires modifying the source code and redeploying the entire application, which can be cumbersome.

Conclusion

The XML editor web application provided valuable hands-on experience in managing XML data dynamically within a web environment. It enhanced our understanding of web development tools and Node.js, reinforcing our skills in handling structured data formats.

Next Steps

Future iterations could focus on enhancing user experience, implementing additional XML manipulation features, and optimizing server-side operations for scalability.

Project Overview:

XML Book List			
Title	Author	Year	Actions
<input type="text" value="The Great Gatsby"/>	<input type="text" value="F. Scott Fitzgerald"/>	<input type="text" value="1925"/>	<input type="button" value="Delete"/>
<input type="text" value="1984"/>	<input type="text" value="George Orwell"/>	<input type="text" value="1949"/>	<input type="button" value="Delete"/>
<input type="text" value="To Kill a Mockingbird"/>	<input type="text" value="Harper Lee"/>	<input type="text" value="1960"/>	<input type="button" value="Delete"/>

Update and click save: To update in XML

XML Book List

Title	Author	Year	Actions
<input type="text" value="The Great Gatsby"/>	<input type="text" value="Akash Varma"/>	<input type="text" value="1925"/>	<input type="button" value="Delete"/>
<input type="text" value="1984"/>	<input type="text" value="George Orwell"/>	<input type="text" value="1949"/>	<input type="button" value="Delete"/>
<input type="text" value="To Kill a Mockingbird"/>	<input type="text" value="Harper Lee"/>	<input type="text" value="1960"/>	<input type="button" value="Delete"/>

PROTOTYPE 2:FINAL PROJECT

IMPORTANT: - PLEASE OPEN THE BELOW LINKS

Detailed Code and explanation: -

<https://docs.google.com/document/d/141pWcgYXQh4WmKjmoJnYgNuV7urgF64z/edit?usp=sharing&oid=106739015895971129567&rtpof=true&sd=true>

Folder Link:

Wetransfer- <https://we.tl/t-XtpNypWJDw>

Drive- https://drive.google.com/drive/folders/1fd6YcW1btBaP3YG7Lgy8l1jJS_C3SRlR

XML Data Editor (Dynamically)

For our final project, we developed an XML Data Editor application using Node.js and various web technologies. This application facilitates the viewing, editing, and saving of XML data through a user-friendly web interface.

Project Overview

Project Directory Structure:

```
xml-editor-app/  
├── index.html  
├── styles.css  
├── script.js  
└── server.js
```

Technologies Used

- **Frontend:** HTML, CSS, JavaScript (ES6+)
- **Backend:** Node.js, Express.js
- **External Libraries:** body-parser for parsing request bodies

Detailed Features

- **File Upload:** Implemented using `<input type="file">` to select and read XML files.
- **Dynamic Display:** Utilized DOM manipulation to generate HTML tables based on XML structure.

- **Data Editing:** Enabled `<td contenteditable="true">` for inline editing of table cells.
- **Adding Data:** Modal dialog for adding new XML data with form inputs dynamically generated.
- **Saving Changes:** POST request to server endpoint (`/save-xml`) to update and save XML data.
- **Styling and Usability:** Responsive design, structured layout, and intuitive user interactions.

Learning and Development

- **Skills Acquired**
 - Gained knowledge in various command and control systems.
 - Developed proficiency in using XML tools in a Linux environment, including jQuery, VS Code, Eclipse, and Atom.
- **Training Sessions**
 - Attended guidance and training sessions on xml data manipulation techniques.

Challenges and Solutions

- Faced challenges for creating accurate records using xml ,jQuery was new topic for our team and we were new to the linux environment during the exploration and testing phases.
- Overcame these challenges by guidance of our mentor mentor Smt Kanan Bala Sahoo.She suggested us the books and clarified our doubts whenever we need it.

Reasons to Move to a Dynamic XML Editor

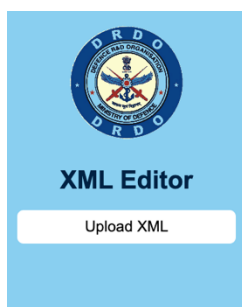
1. **Enhanced Interactivity:**
 - Dynamic applications can provide a richer user experience with features like auto-saving, real-time collaboration, and dynamic content updates without page reloads.
2. **Scalability and Flexibility:**
 - Dynamic applications can better handle large and complex datasets, offering more robust solutions for data manipulation and visualization.
3. **Real-Time Data Handling:**
 - Dynamic editors can support real-time updates, reflecting changes instantly and synchronizing data across multiple clients.
4. **User Personalization:**

- They can store user-specific settings and preferences, offering a customized experience based on user roles or past interactions.
- 5. Improved Maintenance and Upgradability:**
- Dynamic applications often separate the frontend from the backend, making it easier to update, maintain, and scale different parts of the application independently.
- 6. Advanced Features:**
- Features like validation, autocomplete, syntax highlighting, and error detection can be implemented more effectively in a dynamic environment.
- 7. Better Integration with Other Services:**
- Dynamic applications can more easily integrate with other systems and APIs, allowing for seamless data flow and extended functionalities.

Moving to a dynamic XML editor provides more advanced, user-friendly, and scalable solutions, catering to a wider range of use cases and improving overall efficiency and user satisfaction.

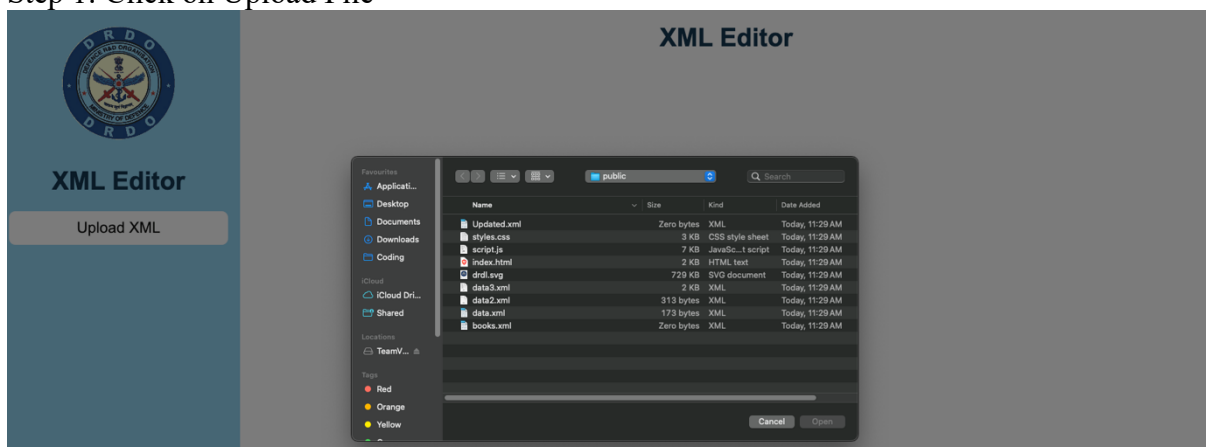
Overview and explanation :

Opening Page




XML Editor

Step 1: Click on Upload File



Step 2:The File Displayed in form of table.



XML Editor

Upload XML

Add New Data


Save Changes

XML Editor

Name	Type	Location	Range	Frequency	Power	Status	Manufacturer	DeploymentDate	Capabilities	
Radar A1	Surveillance	Sector 7G	350	X-band	100	Active	RadarTech	2023-01-15	3D imaging, target tracking	<button>Delete</button>
Radar B2	Tracking	Sector 3B	500	S-band	150	Maintenance	DefenseCorp	2021-05-20	target tracking	<button>Delete</button>

name	capacity	location	status	manufacturer	deploymentDate	supportedVehicles	dimensions	fuelType	safetyFeatures	
Launch Pad Alpha	Heavy	Base 1	Active	SpaceBuild Inc.	2022-03-10	Rockets, Missiles	50m x 50m	Liquid	Blast shields, Fire suppression systems	<button>Delete</button>
Launch Pad Beta	Medium	Base 2	Under Construction	Aerospace Ltd.	2023-08-01	Rockets	40m x 40m	Solid	Fire suppression systems	<button>Delete</button>

Step 3:Alter the data by clicking in box:



XML Editor

Upload XML

Add New Data


Save Changes

XML Editor

Name	Type	Location	Range	Frequency	Power	Status	Manufacturer	DeploymentDate	Capabilities	
Radar A1	Surveillance	Sector 7G	350	X-band	100	Active	RadarTech	2023-01-15	3D imaging, target tracking	<button>Delete</button>
Radar B2	Tracking	Sector 3B	500	S-band	150	Maintenance	DefenseCorp	2021-05-20	target tracking	<button>Delete</button>

name	capacity	location	status	manufacturer	deploymentDate	supportedVehicles	dimensions	fuelType	safetyFeatures	
Launch Pad Alpha	Heavy	Base 1	Active	SpaceBuild Inc.	2022-03-10	Rockets, Missiles	50m x 50m	Liquid	Blast shields, Fire suppression systems	<button>Delete</button>
Launch Pad Beta	Medium	Base 2	Under Construction	Aerospace Ltd.	2023-08-01	Rockets	40m x 40m	Solid	Fire suppression systems	<button>Delete</button>

Step 4:Then click save changes to change the file



XML Editor

Upload XML

Add New Data

Save Changes


XML Editor

Name	Type	Location	Range	Frequency	Power	Status	Manufacturer	DeploymentDate	Capabilities	
Radar A1	Surveillance	Sector 7G	350	X-band	100	Active	RadarTech	2023-01-15	3D imaging, target tracking	<button>Delete</button>
Radar B2	Tracking	Sector 3B	500	S-band	150	Maintenance	DefenseCorp	2021-05-20	target tracking	<button>Delete</button>

name	capacity	location	status	manufacturer	deploymentDate	supportedVehicles	dimensions	fuelType	safetyFeatures	
Launch Pad Alpha	Heavy	Base 1	Active	SpaceBuild Inc.	2022-03-10	Rockets, Missiles	50m x 50m	Liquid	Blast shields, Fire suppression systems	<button>Delete</button>
Launch Pad Beta	Medium	Base 2	Under Construction	Aerospace Ltd.	2023-08-01	Rockets	40m x 40m	Solid	Fire suppression systems	<button>Delete</button>

localhost:3000 says
 XML saved successfully
OK

Step 5: Add new data enables to add new row:



XML Editor

Upload XML

Add New Data

Save Changes

XML Editor

Add New Data

✕

Name

Type

Location

Range

Frequency

Power

Status

Manufacturer

DeploymentDate

Capabilities

Submit

Tracking Delete

res Delete

res Delete

vision Delete

vision Delete

Step 6: Delete enables to delete the existing row:



XML Editor

Upload XML

Add New Data

Save Changes

XML Editor

Name	Type	Location	Range	Frequency	Power	Status	Manufacturer	DeploymentDate	Capabilities	
Radar A1	Surveillance	Sector 7G	350	X-band	100	Active	RadarTech	2023-01-15	3D imaging, target tracking	Delete
Radar B2	Tracking	Sector 3B	500	S-band	150	Maintenance	DefenseCorp	2021-05-20	target tracking	Delete
Radar A4	Tracking	Gitam	22							Delete

name	capacity	location	status	manufacturer	deploymentDate	supportedVehicles	dimensions	fuelType	safetyFeatures	
Launch Pad Alpha	Heavy	Base 1	Active	SpaceBuild Inc.	2022-03-10	Rockets, Missiles	50m x 50m	Liquid	Blast shields, Fire suppression systems	Delete
Launch Pad Beta	Medium	Base 2	Under Construction	Aerospace Ltd.	2023-08-01	Rockets	40m x 40m	Solid	Fire suppression systems	Delete

Tracking Delete

res Delete

res Delete

vision Delete

vision Delete

Step 7: Save Enables to save the changes in data.xml file.

Environment Variables:

Windows Setup and Run Instructions

1. Install Visual Studio Code:

Download and install **Visual Studio Code** from the Official website.

2. Install Node.js:

Download and install Node.js from the [official website](#). This will also install npm, which is Node's package manager.

3. Install Git:

Download and install Git from [git-scm.com](#). This is necessary for cloning the project repository.

4. Clone the Repository:

Use Git Bash or Command Prompt to clone the project repository. For example, use the command:

```
bash
Copy code
git clone https://github.com/your-username/your-repo.git
```

5. Install Dependencies:

Navigate to the project directory in Command Prompt: Copy code

```
cd path\to\project-directory
Install the necessary packages by running:
bash
Copy code
npm install
```

6. Start the Server:

Run the server by executing:

```
bash
Copy code
npm start
```

7. Access the Application:

Open a web browser and go to <http://localhost:3000> to view and interact with the application.

Mac Setup and Run Instructions

1. Install Visual Studio Code:

Download and install **Visual Studio Code** from the Official website.

2. Install Homebrew (Optional but recommended):

Homebrew is a package manager for macOS. Install it by running the following command in Terminal:

```
bash
```

Copy code
/bin/bash -c "\$(curl -fsSL
<https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh>)"

3. **Install Node.js:**

If using Homebrew, run:

bash
Copy code
brew install node

Alternatively, download and install Node.js from the [official website](#).

4. **Install Git:**

Use Homebrew or download from git-scm.com. With Homebrew, use:

bash
Copy code
brew install git

5. **Clone the Repository:**

Clone the project repository using Terminal:

bash
Copy code
git clone https://github.com/your-username/your-repo.git

6. **Install Dependencies:**

Navigate to the project directory:

bash
Copy code
cd /path/to/project-directory

Install the necessary packages by running:

bash
Copy code
npm install

7. **Start the Server:**

Run the server by executing:

bash
Copy code
npm start

8. **Access the Application:**

Open a web browser and go to <http://localhost:3000> to use the application.

Linux Setup and Run Instructions

1. **Install Node.js and npm:**

The method may vary depending on your Linux distribution. For Debian-based systems (like Ubuntu), you can use:

```
bash
Copy code
sudo apt install nodejs npm
```

For other distributions, consult your package manager's documentation.

2. **Install Git:**

Use your package manager to install Git. For Debian-based systems:

```
bash
Copy code
sudo apt install git
```

3. **Clone the Repository:**

Clone the project repository using Terminal:

```
bash
Copy code
git clone https://github.com/your-username/your-repo.git
```

4. **Install Dependencies:**

Navigate to the project directory:

```
bash
Copy code
cd /path/to/project-directory
```

Install the necessary packages by running:

```
bash
Copy code
npm install
```

5. **Start the Server:**

Run the server by executing:

```
bash
Copy code
npm start
```

6. **Access the Application:**

Open a web browser and go to <http://localhost:3000> to interact with the application.

Conclusion

Our summer internship project focused on developing an editable webpage for managing XML-based tables. This experience provided us with valuable skills in web development and XML data manipulation.

We began by exploring various command and control systems to broaden our knowledge. We then tested multiple XML tools in a Linux environment, including jQuery, VS Code, Eclipse, and Atom. This allowed us to identify the most suitable tools for our needs.

The project's core objective was to create a user-friendly webpage for efficient XML table content editing. Using XML, JavaScript, and Node.js, we developed a dynamic interface that simplifies managing XML data. The final product is a robust web application that allows users to view, edit, delete, and save XML data seamlessly, improving user experience and productivity.

We received invaluable guidance from our mentor, Smt Kanan Bala Sahoo, and support from the DRDO team. Our colleagues and friends also provided constructive feedback, enhancing the project's quality.

Future iterations of the XML editor web application could focus on adding more features and improving scalability. Enhancing user experience and implementing advanced XML functionalities are potential next steps.

Overall, this internship has been a transformative experience, equipping us with essential web development skills. We are grateful for the opportunity and look forward to applying what we have learned in future endeavours.