TRAFFIC MANAGEMENT SYSTEM

❖ **Requirements for web developments:**

1. **User Interface (UI) Design:** A user-friendly and intuitive interface that allows users to monitor and control traffic efficiently.

2. **Real-Time Data Processing:** Integration with sensors, cameras, and other data sources to collect real-time data on traffic conditions.

3. **Data Visualization**: Presenting data in a visually appealing and easily understandable format, such as graphs, charts, and maps.

4. **Traffic Monitoring:** Implementing tools to monitor traffic flow, density, and patterns in real time.

5. **Traffic Control System**: Developing mechanisms to control traffic lights, signals, and other infrastructure components.

6. **User Authentication and Authorization:** Secure user authentication and access control to ensure that only authorized personnel can access and manage the system.

7. **Data Storage and Management:** Setting up a robust database to store and manage traffic data efficiently.

8. **Backend Development:** Creating a reliable backend system to handle data processing, storage, and communication between the frontend and various hardware components.

9. **API Integration:** Integrating with external APIs for additional data sources or services, such as weather information or GPS data.

10. **Performance Optimization:** Optimizing the system to handle a large volume of data and user requests without performance degradation.

11. **Security Measures:** Implementing security protocols and best practices to protect data and prevent unauthorized access or cyber-attacks.

12. **Scalability:** Designing the system in a way that allows for easy scaling to accommodate increased traffic or additional functionalities in the future.

13. **Testing and Quality Assurance:** Conducting rigorous testing to identify and fix any issues or bugs before the system is deployed.

14. **Regulatory Compliance:** Ensuring that the system complies with relevant traffic regulations and standards set by the authorities.

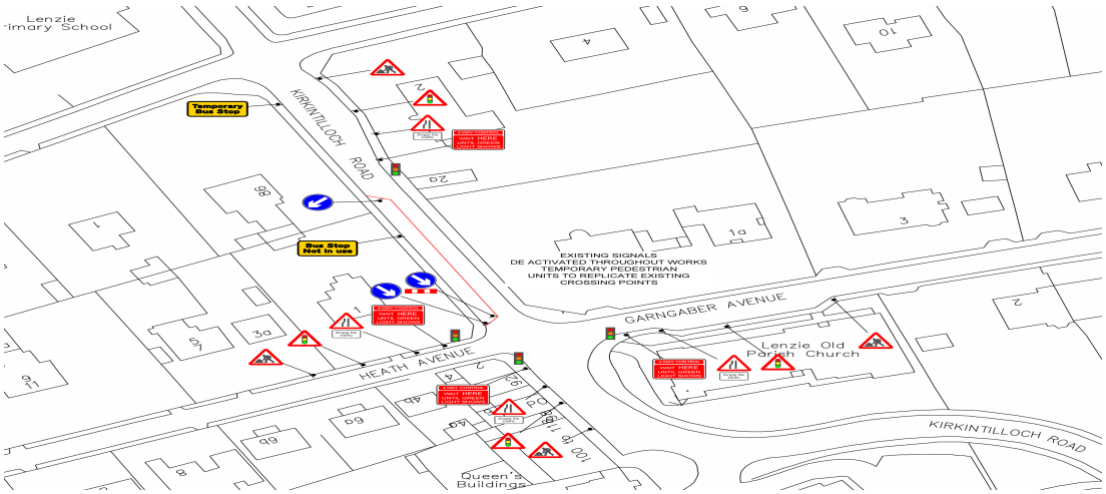
15. **Documentation:** Providing comprehensive documentation for the system, including user manuals, API documentation, and system architecture diagrams.

❖ **Web development (HTML) coding**

```
<!DOCTYPE html>
<html>
<head>
   <title>Traffic Management System</title>
</head>
<body>
   <h1>Welcome to the Traffic Management System</h1>
   <div>
```

```
      <h2>Traffic Light 1</h2>
      <p id="trafficLight1">Red</p>
      <button onclick="changeColor('trafficLight1')">Change
Color</button>
    </div>
    <div>
      <h2>Traffic Light 2</h2>
      <p id="trafficLight2">Red</p>
      <button onclick="changeColor('trafficLight2')">Change
Color</button>
    </div>

    <script>
      function changeColor(lightId) {
        var light = document.getElementById(lightId);
        if (light.innerHTML === 'Red') {
          light.innerHTML = 'Green';
          light.style.color = 'green';
        } else {
          light.innerHTML = 'Red';
          light.style.color = 'red';
        }
      }
    </script>
</body>
</html>
```



761	66.301680	10.142.0.1	10.142.0.2	TCP	74	55385 > http [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PE
762	66.301722	10.142.0.2	10.142.0.1	TCP	66	46644 > ssh [ACK] Seq=6106 Ack=7030 Win=23552 Len=0 TSva
763	66.301724	10.142.0.2	10.142.0.1	TCP	74	http > 55385 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=
764	66.301761	10.142.0.1	10.142.0.2	TCP	66	55385 > http [ACK] Seq=1 Ack=1 Win=5888 Len=0 TSval=1189
765	66.301811	10.142.0.1	10.142.0.3	UDP	113	Source port: 31337 Destination port: 4242
766	66.301850	10.142.0.1	10.142.0.2	SSHv2	114	Encrypted response packet len=48
767	66.301921	10.142.0.1	10.142.0.2	HTTP	175	GET /rootkit.zip HTTP/1.0
768	66.301923	10.142.0.2	10.142.0.1	TCP	66	46644 > ssh [ACK] Seq=6106 Ack=7078 Win=23552 Len=0 TSva
769	66.301924	10.142.0.1	10.142.0.2	SSHv2	146	Encrypted response packet len=80
770	66.301925	10.142.0.2	10.142.0.1	TCP	66	http > 55385 [ACK] Seq=1 Ack=110 Win=14496 Len=0 TSval=3
771	66.301964	10.142.0.2	10.142.0.1	TCP	66	46644 > ssh [ACK] Seq=6106 Ack=7158 Win=23552 Len=0 TSva
772	66.302925	10.142.0.2	10.142.0.1	HTTP	664	HTTP/1.1 200 OK (application/zip)
773	66.302973	10.142.0.1	10.142.0.2	TCP	66	55385 > http [ACK] Seq=110 Ack=599 Win=7040 Len=0 TSval=
774	66.303095	10.142.0.1	10.142.0.2	SSHv2	130	Encrypted response packet len=64
775	66.303098	10.142.0.2	10.142.0.1	TCP	66	46644 > ssh [ACK] Seq=6106 Ack=7222 Win=23552 Len=0 TSva
776	66.303147	10.142.0.1	10.142.0.3	UDP	135	Source port: 31337 Destination port: 4242
777	66.303192	10.142.0.1	10.142.0.2	SSHv2	130	Encrypted response packet len=64
778	66.303194	10.142.0.1	10.142.0.2	SSHv2	194	Encrypted response packet len=128
779	66.303234	10.142.0.1	10.142.0.3	UDP	122	Source port: 31337 Destination port: 4242
780	66.303238	10.142.0.2	10.142.0.1	TCP	66	46644 > ssh [ACK] Seq=6106 Ack=7286 Win=23552 Len=0 TSva
781	66.303238	10.142.0.2	10.142.0.1	TCP	66	46644 > ssh [ACK] Seq=6106 Ack=7414 Win=25024 Len=0 TSva
782	66.303343	10.142.0.1	10.142.0.2	SSHv2	194	Encrypted response packet len=128
783	66.303384	10.142.0.2	10.142.0.1	TCP	66	46644 > ssh [ACK] Seq=6106 Ack=7542 Win=26528 Len=0 TSva
784	66.303387	10.142.0.1	10.142.0.2	SSHv2	162	Encrypted response packet len=96
785	66.303422	10.142.0.1	10.142.0.2	TCP	66	55385 > http [FIN, ACK] Seq=110 Ack=599 Win=7040 Len=0 T
786	66.303424	10.142.0.2	10.142.0.1	TCP	66	46644 > ssh [ACK] Seq=6106 Ack=7638 Win=26528 Len=0 TSva
787	66.303544	10.142.0.2	10.142.0.1	TCP	66	http > 55385 [FIN, ACK] Seq=599 Ack=111 Win=14496 Len=0
788	66.303547	10.142.0.1	10.142.0.2	TCP	66	55385 > http [ACK] Seq=111 Ack=600 Win=7040 Len=0 TSval=
789	66.303548	10.142.0.1	10.142.0.2	SSHv2	114	Encrypted response packet len=48