Project: Smart Traffic Light System (STLS)
Members: Thomas Bonagura, Liam Murphy, Courtney Moane, Bhaskara Yashwant Bitra

1.1	Traffic Flow AI Algorithms	To develop AI for managing traffic, we start by collecting a lot of traffic data. We use Python, with libraries like Pandas and NumPy, to clean and prepare the data. This step makes sure that our machine learning models, which we build using TensorFlow and PyTorch, get good-quality data to work with. Our main goals are to predict traffic flow and make traffic lights smarter by adjusting them in real time. We plan to use time-series forecasting to predict traffic and Q-learning to make traffic lights more adaptive. To test our models, we'll use a program called SUMO (Simulation of Urban MObility). This package lets us run tests in different situations before we use the models in production. Once everything is tested, we will connect the AI models to the traffic control system through RESTful APIs. This way, the AI can easily communicate with the traffic lights and other hardware devices. We plan to also use Docker containers to help us deploy our models on servers easily, making sure we can scale up without too much trouble.
1.2	IoT Traffic Sensors Installation	We will start by installing IoT traffic sensors in places we have determined to be best suited for traffic optimization using traffic data from ArcGIS. By doing this we enhance our coverage and improve our data. We plan to use a multimodal traffic monitoring platform with mmWave Radar beams to track traffic in real-time. These sensors will be connected through using available 5G networks to rapidly transfer data to our main traffic server allowing for large-scale, real-time traffic optimization decisions. We will ensure the sensors are installed up to code to preserve their longevity. Before we deploy, we will run comprehensive testing to ensure the data and network are reliable, using tools that mimic different traffic scenarios. This can include JUnit testing and other forms of unit tests depending on the languages we use. If tests pass, we'll continue to the next phase, where the system is officially up and running.
1.3	Central Traffic Management Server	A strong server system is essential for setting up our traffic management center. We are going with 10 top-notch Dell PowerEdge servers. They have the latest Intel Xeon CPUs to handle data super fast and run on Linux with Apache for the best security and reliability. This setup is meant to deal with huge amounts of data, up to 50TB, from all over the city's traffic sensors. Our servers will not just analyze traffic with our AI software but also manage all the data from the IoT sensors, making real-time traffic decisions possible. We plan on regular maintenance to keep everything running smoothly, including updates and check-ups to ensure our system stays top-notch.
1.4	Sensor and Camera Data Integration API	To make sure our IoT sensors and traffic cameras work effectively with our main traffic system, we are building an industry-grade API using the Django REST framework. We chose it because it's great at dealing with complicated data and works well with Python, which we use for all our backend material. This project is all about making a middle layer that can quickly send data from all the traffic sensors and cameras in the city to our server. This lets us make traffic decisions on the fly based on current information. We are focusing on making this API extremely fast, using async programming and caching so we can change traffic signals in short periods

		of time. We plan to test this API multiple times, using tools like Postman and LoadRunner, to make sure it can handle a lot of data when the city's traffic is at its busiest.
1.5	Traffic Data Database	We are setting up an enterprise-level database to keep track of all kinds of traffic data, including real-time data. Including traffic levels on the roads, accidents, and traffic flows at different times and locations. We are using MySQL due to its scalability and reliability. This database is key for making quick traffic decisions and for planning how to improve the city's roads over time with the use of efficient query processing. We are giving a large amount of attention to security, encrypting stored data (with AES-256) and data being sent around (with SSL/TLS), and we will dynamically check for any security issues. Setting up this entity means getting servers ready, planning out the database structure and schema, and making sure we can quickly find and store data. This is crucial for keeping our traffic system fast and reliable.
1.6	Public Reporting Mobile App	Our project focuses on giving the community a role in managing traffic by letting people use a mobile app to report traffic issues and see what's happening on the roads in real-time. We are building this app for both iOS and Android, using React Native to make sure it works well on both platforms. The app will have features like GPS to share exact locations, a connection with the central traffic system to exchange data in real-time, and alerts to keep everyone updated on traffic conditions. We will be doing many tests to make the app easy to use and reliable. Our goal is to get the community involved, which will help optimize traffic management through the use of crowd-sourced data from the people who are on the roads.
1.7	Traffic Light Hardware Upgrade	We are planning to update our traffic lights to work more effectively with our new traffic management system. This means putting in new smart controllers and communication components that can communicate and distribute information with our new AI system without any problems. We are going to upgrade traffic lights to make them more flexible and able to change based on what commuters report in the system. The upgrade will involve adding new hardware that lets the traffic lights understand and follow the AI system's instructions. We plan to purchase the new compatible hardware, install it, and then test it thoroughly to make sure it runs smoothly without causing any issues for drivers. This upgrade is a big step towards modernizing our traffic control system to adjust to traffic in real-time, providing drivers with a safer and smoother experience.
1.8	User and Administrative Web Interface	We are working on creating web interfaces for both regular users and system administrators, making it easier for everyone to interact with traffic data and controls. The public side of the system will let people see traffic in real time, report issues, and get alerts on what is going on around them. For administrators, there will be a dashboard to access a more in-depth look into traffic patterns, managing traffic signals, and watching the AI system. We will use React for a smooth user experience and Node.js for a strong backend. The admin side will also use GIS tech for detailed maps and planning. We plan to put a great amount of focus on security, with things like two-factor authentication and controls on who can access certain information and to keep the data safe.

1.9	System Security and Data Protection	This stage is about building a solid security system for our traffic management. We are setting up top-notch encryption to keep data safe, strong firewalls to block cyber threats, and performing many tests to find any weak spots. We also plan to work with cybersecurity professionals to make sure we are doing everything right and staying up to date with security standards. Training our team on cybersecurity and legal importances is another large part of this, as is starting a security operations center (SOC) to watch out for any threats. This allows us to create the most optimally secure platform that is safe for people to use on a daily basis.