### User Stories for Managers:

As a manager, I want to assess information on the Cab Driver’s car models so that I can tell when older car models are in use which might be more prone to accidents so I can proactively plan for car upgrades and improve fleet safety.

**Confirmation (Dashboard/Python)**:

1. **Pie Chart (Dashboard)**: Show the composition of the fleet by car brand, highlighting the percentage of each brand in the fleet.
2. **Bar Graph (Dashboard)**: Illustrate the number of bookings for each car made, alongside the proportion of safety warnings associated with each brand to visualize how often specific car brands are used and whether they result in safe trips or not.
3. **Tree Maps (Dashboard)**: Depict the average age of the car brands, based on their manufacturing year.
4. **Countplot (Python)**: Depict bins of car age against the percentage of unsafe trips per driver to see if the year when the car was manufactured affects how safe the trip will be.

As a manager, I need to access telematics data (e.g., acceleration, speed) to evaluate drivers' behavior and identify drivers who may need additional training or coaching to enhance safety. Additionally, I use this data to see which driver is our top earner and which driver could be a potential risk to our operations. This comprehensive analysis is crucial for ensuring both safety and efficiency in our team.

**Confirmation (Dashboard)**:

1. **Bar Graph (Dashboard)**: Identify the top 10 cab drivers with the highest number of recorded dangerous trips to point out drivers who might pose a danger to the riders.
2. **Bar Graph (Dashboard):** Highlight the top 10 cab drivers with the most bookings and analyze if they are safe or not using filters.
3. **Driver KPI Analysis (Dashboard)**: Evaluate drivers statistics and safety labels to determine their compliance with safety standards a (based on their no. of bookings and ratings)
4. **Violin Plots (Python)**: Check the distribution on of each safety sensor (i.e. Gyro\_x, Gyro\_y) to visualize the proportion of drivers that surpass a certain limit.

As a manager, I need an intuitive application with a user interface so that I can visualize, analyze and compare cab-driving data such as the telematics data on trip safety and make informed decisions. (In the CA1 Brief)

**Confirmation (Dashboard/GUI/Machine Learning)**:

1. **Personalized Graphs (Dashboard)**: Include analytical features or tools like asking for user inputs to allow the managers to derive personalized insights based on their requirements.
2. **Driver Statistics KPI (Dashboard):** Application should allow managers visualise whether the drivers have hit their number of trips quota for the day based on the threshold levels the managers entered.
3. **Safety Statistics KPI (Dashboard)**: Application should enable the manager to compare different aspects of cab driving data, such as safety metrics (i.e. Speed, gyro) among various drivers.
4. **UI Friendly Webpage (GUI)**: Applications should be designed user-friendly and easily navigable, allowing the managers to have personalized inputs to ensure a more positive user experience for the manager.
5. **Machine Learning Prediction (Machine Learning)**: Application should make use of machine learning to determine if the cab driver's particular trip is a safety risk or not.

### User Stories for Administrators:

As an administrator, I want to manage user roles and permissions, so that only authorized personnel have access to sensitive data within the application, enhancing data security and privacy.

**Confirmation (APIs/Security)**:

1. **Access Control Requirement (APIs)**: Allow the administrator to define and manage user roles, ensuring that authorized personnel can access specific data.
2. **Data Security Focus (Security)**: The application should prioritize security measures, including password hashing, secure authentication to safeguard sensitive information.

As an administrator, I need to establish data storage and backup procedures so that data integrity and availability is ensured, reducing the risk of data loss and supporting disaster recovery.

**Confirmation (SQL/SQL Alchemy)**:

1. **Data Integrity Priority (SQL)**: Use secure databases like MySQL Server to maintain and consistency of stored data, and procedures for regular data backups.
2. **Data Availability Requirement (SQL)**: Application should have provisions for continuous data availability, minimizing downtime and ensuring uninterrupted access to critical information from the databases.
3. **ETL Pipeline (SQL Alchemy)**: Ensure that the necessary data can be retrieved from the database securely without any data leakage.

### User Stories for Cab Drivers:

As a cab driver, I want my personal information to be shown to the customers such as my years of experience, gender so that they know they are receiving the safest and most comforting treatment.

**Confirmation (Dashboard/GUI/Python)**:

1. **Driver Ratings KPIs (Dashboard)**: Display the drivers’ name, their ratings, gender, year of experience, drivers’ age, car make, age of car manufactured.
2. **Driver Ratings VS Gender Bar Graph (Python)**: Display the statistics of the driver ratings in bins among different cab drivers’ gender for drivers to see if gender stereotypes affect how customers rate them.
3. **Chi-Square Contingency Test (Python)**: Use the Chi-Square Test for a more accurate analysis to determine if gender will affect the driver ratings.
4. **User Friendly Interface** **(GUI):** Ensure that UI is user friendly and UX is smooth by ensuring that customers are able to view their respective drivers’ necessary information in a clear table manner.

As a cab driver, I want my personal performance (ratings and sensor metrics) to be analysed and monitored, whether I am providing my riders with a safe trip so that customers will be assured of my competence and feel safe during the trip.

**Confirmation (Dashboard/Python)**:

1. **Driver Safety KPIs (Dashboard)**: Display the driver’s percentage of safe trips and the counts of the booking that were safe.
2. **Bookings Safety KPIs** **(Dashboard)**: Display a KPI for the booking that shows whether the trips are safe or unsafe generally.
3. **Sensors Data Graphs (Dashboard)**: Display graphs (Speed, Acceleration, Gyro) to show at which part of the trips reveals any safety risks from checking whether the safety metrics surpassed a certain threshold level.
4. **Histogram (Python)**: Display a histogram to visualise what the distribution is like for the number of trips per driver so I can see if I am performing as well as the other drivers or underperforming.
5. **Barchart (Python)**: Display the frequency of booking trips safety labels to visualise if they are generally considered safe or unsafe.
6. **Barchart (Python)**: Display the average driver ratings grouped by car brands to check if the car models affect the number of ratings given by the customers as certain car brands is able to provide customers a more comfortable journey.
7. **Clustering (Python)**: With the use of clustering, determine whether the driver falls in a certain cluster that require additional feedback and service quality training. The different clusters can also provide additional insight on whether the drivers’ traits like age/years of experience cause a higher number of ratings.