**Auto-Tracker Presentation Script**

**Slide 1:** Our team consisted of Richard Ay, Owais Islam, Lorena Kell, and Cameron Wills

**Slide 2: Elevator Pitch** – There are families and organizations with multiple vehicles, each with different time-sensitive data that must be maintained. Registration and insurance must be updated annually or semi-annually, tire and oil change mileage must be tracked, VIN and toll tag numbers must be available for various purposes.

Failure to track and update this information can lead to fines and/or unsafe driving conditions. The Auto-Tracker application enables efficient storage and access to this information for any web-enabled device, eliminating the needs for paper notes scattered around which are easy to lose or overlook.

**Slide 3: The User Story** – As an owner of several vehicles, I want an application that can track: registration renewal dates, insurance expiration dates, VIN numbers, toll tag numbers, the purchase mileage, the mileage of the last tire change, the mileage of the last oil change, and the name of the primary driver. From the application I want to login and see my data, then be able to add or update drivers, and add or update vehicles. The application should be accessible from any web-enabled device, and be responsive to each form factor I use.

**Slide 4: Application Concept** – The Auto-Tracker application is based on the MVC architecture, and stores its information in a SQL database. The application takes advantage of ‘sequelize’ for ‘object relational mapping’ to the database routes. Additionally, the AWS S3 npm package is utilized to store an image of each vehicle. Complete CRUD functionality is provided for the maintenance of the data.

**Slide 5: Processes, Technologies, Challenges, Successes** – A number of NPM packages were used in this application to achieve responsiveness, authentication, code modularization and database operations. The team members split up the work, as time and schedules permitted, focusing on the “next important issue to be resolved”. Much of the actual coding occurred during Zoom sessions in groups of two or three.   
  
The challenges encountered while developing this application include: determining which modal was activated, manipulating images, login and sign-up authorizations. Successes included peer programming and finally a working application.

**Slide 6: Application Details** – The expected usage/workflow through the application is:

* The user/owner logs in (or signs-up), and then has access to selected vehicles based on the ‘owner\_id’.
* The owner can then define a number of drivers.
* The owner can then define a number of vehicles, associating them with drivers and defining the pertinent details of each vehicle.
* When necessary, the data for drivers and/or vehicles can be updated as necessary.

**Slide 7: Application Demonstration** – Run the application from Heroku

* Log in and show the vehicle and driver dashboards.
* Add a vehicle
* Show the details of a vehicle.
* Edit the details of a vehicle.
* Delete a vehicle and review the vehicle dashboard again.

**Slide 8: Future Development** – Ideas for future development include:

* Use a commercial API to acquire an exact image of each vehicle based on VIN.
* Use a commercial API to acquire more vehicle specific data for each vehicle.
* Improve the ‘add vehicle’ UI to show a list of current drivers.

**Slide 9: Important Links** – This slide just shows the (required) submission links for Heroku and GitHub.

**Slide 10: Questions** – Let’s see what comes back.