a) This web application is a simple inventory management system for parts, using their part ID, name, and amount. It allows users to perform CRUD (Create, Read, Update, and Delete) operations on the database that contains the part information. Users can add new parts, edit the quantity of existing parts, delete parts, and view all parts in the database. To add, users can provide the part ID, name, and amount. If the part ID already exists, it will add the provided amount to the already existing part. To read, users can view all parts and other information in the database in a table format by pressing the “View Parts” button. To update, users can specify a part ID and a new amount. To delete, the user can give a part ID to remove that part from the database. The user also has the option to clear the entire table. To run the app, you navigate to the file location in a command-line interface, and run “python -m flask run”.

b) The backend is built using Python with a Flask framework, and SQLite is used as the database to store part information. The database stores the id (integer), name (text), and amount (int) of a part. The app handles the CRUD operations with Flask routes (/add, /edit, and /delete). The app also contains basic error handling, requiring the user to enter integers for part IDs and amounts.

c) In the past, mainly during robotics, there have been difficulties communicating parts that need to be bought, often buying too many or too little. However, with this app, being able to add the part and amount I need to a database that everyone shares has made communicating which parts are required much easier. For example, when I needed the coach to buy 3 more 80 tooth gears, I just added them to the website and the coach bought it and then removed it from the database. Having a shared inventory has made communication about buying parts much smoother.