# CHATLOG.md

This is a log of my work with ChatGPT while building Assignment 1. I asked questions about class design, dataset handling, and debugging. ChatGPT gave me suggestions and small code pieces. I wrote and tested the final project myself.

Me: The assignment says to make a `Car` class. Should I just use the attributes in the instructions?

ChatGPT: Yes — add `manufacturer`, `model`, `year`, `mileage`, `engine`, `transmission`, `drivetrain`, `mpg`, `exterior\_color`, `interior\_color`, `accident`, and `price`.

Me: Here’s my first attempt at the constructor:

class Car:  
 def \_\_init\_\_(self, manufacturer, model, year, mileage):  
 self.manufacturer = manufacturer  
 self.model = model  
 self.year = year  
 self.mileage = mileage

ChatGPT: That’s a good start, but you still need the rest of the attributes like `engine`, `transmission`, `drivetrain`, colors, accident history, and price. Don’t forget to make `mpg` optional.

Me: Okay, updated. What about the `Modify\_Price` method?

ChatGPT: If the input value is >= 1, set the price directly. If it’s less than 1, treat it as a discount. For example:  
  
def Modify\_Price(self, value: float) -> None:  
 value = float(value)  
 if value >= 1:  
 self.price = round(value, 2)  
 else:  
 new\_price = round(self.price \* (1 - value), 2)  
 print(f"Price modified from ${self.price} to ${new\_price}.")  
 confirm = input("Apply this new price? (y/n): ")  
 if confirm.lower() == "y":  
 self.price = new\_price

Me: Now for the `Seller` class. I have this:

class Seller:  
 def \_\_init\_\_(self, name):  
 self.name = name  
 self.inventory = []

ChatGPT: Add a `rating` attribute with a default (like 4.5). Then create `Buy` and `Sell` methods. Also, a `list\_brief()` function to quickly print inventory items.

Me: Here’s my `Buy` method:  
  
def Buy(self, car):  
 self.inventory.append(car)

ChatGPT: Works, but add a check to make sure you don’t buy the same car twice.

Me: For `Sell`:  
  
def Sell(self, car):  
 self.inventory.remove(car)

ChatGPT: Add error handling: if the car isn’t in the inventory, raise a `ValueError`.

Me: Now I’m stuck on reading the CSV.

ChatGPT: Use `csv.DictReader`. Create a `MAP` dict that connects common column names to your fields, like `make` → `manufacturer`, `odometer` → `mileage`.

Me: I wrote `\_to\_float` but it sometimes fails when mileage is blank.

ChatGPT: Add a default return if parsing fails:  
  
def \_to\_float(v, default=None):  
 if v is None or v == "":  
 return default  
 try:  
 return float(v)  
 except:  
 import re  
 m = re.search(r"(-?\d+(\.\d+)?)", str(v))  
 if m:  
 return float(m.group(1))  
 return default

Me: My loader works but sellers all show 0 cars.

ChatGPT: When building sellers, make sure you actually call `s.Buy(car)` inside your loop. That’s why inventories were empty.

Me: Okay fixed. Now `app.py`. I want a menu.

ChatGPT: Use a `while True` loop with options:  
1. List sellers  
2. Show inventory  
3. Modify a car  
4. Move a car between sellers  
5. Quit

Me: I added choice 2, but not 1. Do I need both?

ChatGPT: Yes. `choice == "1"` should list all sellers with their car counts. `choice == "2"` lets you pick a seller and see its cars.

Me: I got an error: `CSV not found at: data/USA\_cars\_dataset.csv`.

ChatGPT: You need to put the CSV inside a `data/` folder in your repo. Or run the script with a path:  
  
python app.py data/USA\_cars\_dataset.csv

Me: It finally works! It loaded 762,000 cars.

ChatGPT: Perfect. For debugging, you can print counts at startup. Add a `DEBUG = True` flag near the top of `app.py`.

Me: Should I put anything in README about the dataset?

ChatGPT: Yes, explain that the dataset comes from Kaggle, link to it, and say it must be placed in `data/USA\_cars\_dataset.csv`.

Me: Okay, I think I’m done. Anything else?

ChatGPT: Add a `CHATLOG.md` file like this one, commit everything, and push to GitHub. Then submit the repo link on Canvas.

# End of Chat  
  
This log shows how I used ChatGPT for guidance. I wrote, tested, and debugged the final solution myself.