1. What challenges did you face when reading an unknown number of words for menu items, and how did you solve them?

The biggest challenge was that menu item names could contain multiple words, so I couldn't just use methods like next() which only grab one word at a time. I solved this by using nextLine(), which reads the entire line and captures all the words in the menu item.

2. What did you learn about Java's ability to get and format the current date and time? How did you figure this out?

I learned that Java makes it really easy to work with dates and times by using classes like LocalDateTime and DateTimeFormatter. The classes let you just grab the current date and time and format it in almost any way you want. I figured it out by checking the Java API documentation and experimenting with how these commands output.

3. How did using parallel arrays impact the way you stored and retrieved menu data? Would another data structure, like a HashMap, have been more efficient?

Using parallel arrays worked well for this assignment because I could store menu names in one array and prices in another, keeping the same index for related items. It was simple and effective for the three-item menu. If there was a lot more data, I think a HashMap would have been better to use

4. What techniques did you use to ensure that the receipt formatting was clear and aligned properly?

I used the printf() function and specified the format so there were fixed widths for each column like 30 characters for the name and 10 for the quantity. This made sure that all of the outputs lined up properly. I also built the receipt using a StringBuilder so I could easily format and adjust everything before printing to the terminal or writing it to a separate file.