*You are CoDev, an expert full-stack programmer & product manager with deep system and application expertise and a very high reputation in developer communities. You are also a master in all computer algorithms and optimisations. You always write code taking into account all failure scenarios and errors. You’ve launched multiple products with optimised user experiences. I’m your manager, and you are expected to write a program, following the commands I’ll instruct. You will always use the latest language features and APIs/packages, and will ensure the syntax is correct to the best of your knowledge and abilities. You will follow the below commands, and will only output the result or code unless you are asked to provide any commentary or descriptions. You can only output filenames, folder structures, code, tests. You can speak only for asking clarification questions. Please ensure the code that you output is valid to the best of your knowledge. If you need clarification, just ask. Below are the commands you should follow along with the related instructions. All commands will be of the format /command [parameter1] [param2] [param3]*

*/project— When you receive this command, output the list of files & folder structure you’ll be having in this project, based on the project summary and task you’ve to accomplish.*

*/code [filename] — When you receive this command, output the code for the file indicated with the filename. This should be a filename that you mentioned after receiving the /project command. Ensure the functions of the file work well in tandem with other files and functions/modules in your project, to accomplish the task indicated.*

*/tests [filename] — When you receive this command, output the tests for the file indicated with the filename*

*/explain [filename] [function] — When you receive this command, explain the given function in the filename*

*/run — When you receive this command, simulate the console of the program when it is running.*

*/revise [filename] [modification] — When you receive this command, re-write the content of the file [filename] ensuring the functions of the file work well in tandem with other files and functions/modules in your project, also by taking the modification into consideration*

*/comment [filename] [function] — Add a comment to the function in the file*

*/format [filename] — Format the code with in the file properly.*

*/help — when you see this, output “CoDev — A GPT 4.0 Virtual Developer” followed by a list all the commands that are possible other than /help, along with few examples and a description*

Create a python web app for tracking my sales on Ebay and Amazon, and tracking the performance of my inventory purchase lots through the sales from Ebay and Amazon.

This app will need at least these models:

* Purchase lot – will be the model to represent my inventory purchases. Will only be created by a user through a form. Will need these attributes:
  + Date of Purchase – Input by user through form on creation or update
  + Lot identifier – Input by user through form on creation or update
  + Venue of Purchase – Input by user through form on creation or update
  + Venue Transaction Number – Input by user through form on creation or update
  + Original Source – Input by user through form on creation or update
  + Type (Online Arbitrage, Retail Arbitrage, Liquidation, Wholesale) – Input by user through form on creation or update
  + Main Category – Input by user through form on creation or update
  + Short Description – Input by user through form on creation or update
  + Units Received – Input by user through form on creation or update
  + Subtotal Cost – Input by user through form on creation or update
  + Fees on Purchase – Input by user through form on creation or update
  + Shipping (inbound) – Input by user through form on creation or update
  + Other Cost – Input by user through form on creation or update
  + Discounts – Input by user through form on creation or update
  + Total Cost – Total of Sum of Subtotal Cost, Fees on Purchase, Shipping (inbound), Other Cost, Discounts
  + Link to webpage – Input by user through form on creation or update
  + Units Sold – Sum of Units Sold of it related Items
  + Gross Revenue - Sum of Units Sold of it related Items
  + Shipping (Outbound) - Sum of Units Sold of it related Items
  + Fees on Sale - Sum of Units Sold of it related Items
  + Tax on Sale - Sum of Units Sold of it related Items
  + Net Revenue – Total of (Gross Revenue – Shipping (Outbound) – Fees on Sale – Tax on Sale)
  + Profit/Loss $ - Total of (Net Revenue – Total Cost)
  + Profit/Loss % - Quotient of (Profit/Loss $ / Total Cost)
* Item – will be related to Purchase Lot (Purchase Lot has many Items). Will represent each individual SKU that I received in a Purchase Lot, may be created by a user through a form or programmatically when a new SKU is recognized through an order line item:
  + SKU identifier – Can be input by a user at record creation or programmatically when a new sku identifier is recognized on an order line item
  + Name / Title – Can be input by user or programmatically on record creation
  + Units Received – will be input by user either during record creation or through update
  + Units Sold – cumulative total through all line item sales
  + Cost Per Unit – Will be input by user on record creation or update
  + Revenue – cumulative total through all line item sales
  + Average Sale Price- Quoitent of (Revenue / Units Sold)
  + Shipping Cost (Outbound) - cumulative total through all line item sales
  + Fees – cumulative total through all line item sales
  + Net Revenue – cumulative total of (Revenue – Shipping Cost (Outbound) – Fees)
  + Profit/Loss $ - cumulative total of (Revenue – (Cost Per Unit \* Units Sold))
  + Profit/Loss % - Quotient of (Profit/Loss $ / (Cost Per Unit \* Units Sold))
* Order – an order will represent an order placed by a customer through sales channel (Amazon, Ebay, etc):
  + Order Date – input programmatically when order is created from API response
  + Order Venue – input programmatically when order is created from API response (if from Ebay = Ebay, if from Amazon = Amazon)
  + Order Number – input programmatically when order is created from API response
  + Subtotal Sale Price – input programmatically when order is created from API response
  + Shipping/Handling Charged – input programmatically when order is created from API response
  + Tax – input programmatically when order is created from API response
  + Total Paid By Buyer – input programmatically when order is created from API response
  + Fees – input programmatically when order is created from API response
  + Shipping Cost – input programmatically when order is created from API response
  + Net Revenue – Total of (Total Paid By Buyer – Fees – Shipping Cost)
* Line Item – will be related to an Order (an Order has many Line Items) and to item (a Line Item has one Item):
  + Line Item Number – input programmatically when line item is created from API response
  + Quantity – input programmatically when line item is created from API response
  + Sale Price – input programmatically when line item is created from API response
  + Shipping/Handling Charged – input programmatically when order is created from API response
  + Total Paid by Buyer – input programmatically when line item is created from API response
  + Tax – input programmatically when line item is created from API response
  + Fees – input programmatically when line item is created from API response
  + Shipping Cost – Will be calculated by dividing the shipping cost of the order by all of the orders line items equally
* Transaction – will be related to an Order (may have one Order) and a line item (may have one line item):
  + Transaction Date – input programmatically when transaction is created from API response
  + Transaction Number – input programmatically when transaction is created from API response
  + Venue - – input programmatically when transaction is created from API response (if from Ebay = Ebay, if from Amazon = Amazon)
  + Transaction Type – input programmatically when transaction is created from API response
  + Transaction Value – input programmatically when transaction is created from API response
  + Booking Entry (debit or credit) – input programmatically when transaction is created from API response

This app will pull Order and line item information from the Ebay and Amazon APIs, load that info into the database, match to any related items, and purchase lots. Pull transaction info from the Ebay and Amazon APIs, load that into the database, and match to any matching orders, line items, items, purchase lots, update any related totals.

I would like the app to be able to pull the information on request, by date range. Every time a the information is pulled from their respective APIs and the database is successfully updated with that information, I would like the date of that pull request to be save, so that the next time the user request to call the APIs, the date range for the request will be since that saved date of the last request.

I would like to have a static navigation bar at the top of the page with drop downs. I would like the page to have a simple, night mode style.

Please create all test for this project as well. If you need any clarification, or need any other input from me, please let me know.