Commerce Planning

1. Functionality

**Create Listing**

Users should be able to create listings. This means that there should be a form to submit the inputs necessary to create a listing

**Active Listings Page**

Users should be able to go here and see all active listings. [Means that there should be a flag on listings to show whether they are active or not]. Should see details such as (i) current price (ii) title (iii) description (iv) photo

Means we would return the listing objects from the table and display them

**Listing Page**

Clicking on a specific listing should take users to a page specific to that listing. Users should see all details about listing. Users should also be able to

* Add item to their watchlist. If already in watchlist should be able to remove it
* Place bid -> Must validate this such that bid is at least as large as starting bid and greater than any other bid
* If is the author of listing, should have ability to close the auction which makes the highest bidder the winner of the auction and make listings no longer active [Means listings should have a field keeping track of who is the current winner]
* If closed listing page, should show who won that page
* Users who are signed in should be able to add comments to the listing page. Listing page should display all comments

**Watchlist**

Should display all the listings that a user has added to their watchlist.

**Categories**

Should be able to visist a page that displays a list of all listing categories [Means we should have a table for categories and then have listings reference to that]

**Django Admin Interface**

View, add, edit and delete any listings, comments and bids

**Price Action Page**

User should be able to see a history of (1) who bid on their items (2) date that bid was placed (3) How many times their item was viewed

1. Data Model
2. User
3. Id
4. Username
5. First\_name
6. Last\_name
7. Email
8. Is\_staff
9. Is\_active
10. Date\_joined
11. Listings
12. Id
13. Is\_active
14. current price
15. title
16. description
17. photo
18. date\_created
19. Category [Foreign key] -> 1 listing to many i.e 1 listing can have 1 category. But 1 category can have multiple listing
20. Winner [can be empty]
21. Creator -> One to many r/s -> 1 listing can only have 1 creator. But 1 creator can create many listings
22. Watchers -> ManyToManyField to represent a watchlist which is basically a many to many r/s between Listings and Watchers. Note, in SQL normally you would have another table to represent this called a join table. Furthermore, you can conceptualize a row representing 1 listing for every other field except this. This is because this field is basically a hidden table. [Better to make it explicit]
23. Bids
24. Id
25. Bidder -> 1 to many r/s: 1 bid can have 1 bidder but 1 bidder can have multiple bids
26. Price -> bid price by user
27. Date
28. Comments
29. Id
30. Author -> 1 to many: 1 comment can have 1 author. 1 author can have multiple comments
31. Categories
32. id
33. Name of category
34. Date\_created
35. Views
36. Objects

User

Listing

Comments

Things to include next time

1. Static Typing -> Hints + linter installation in Python
2. OOP Analysis, Design, Implementation