EasyBid

Goods Bidding and Auction Website

Team 10
Sherry Wu, Haobin Tan, Congtao Kuang

Introduction

- e-Commerce is increasing fast
- More goods purchases will be e-based or web-based in the future (e.g. Amazon, eBay, Wish). Since the start of the Covid-19 pandemic, Amazon profits increased nearly 200%!
- Bidding and auction seems not so popular so far
- We decide to build a **goods bidding and auction website**

UI Design

- Refer to Figma design: https://www.figma.com/file/qF9AHZYetTJViiSvs5Bb4C/EasyBid?node-id=0%3A1
- Our current implementation: http://127.0.0.1:8000/

Websocket

- Biding and auction is real-time sensitive
- Update the current price and goods status in real-time
- We need to implement a fast communication method
- Django-Channels should be working for this task
- Websocket demo - chat room: http://127.0.0.1:8001/

Data Model

- User (current: django default authentication)
    - username
    - first_name
    - last_name
    - email
    - password

* OAuth might be added later (provide different ways for users to register)

- Profile
    - owner (OneToOneField: user, primary key)
    - bio_text (CharField)
    - profile_picture (FileField)
    - content_type (Charfield)

Data Model

- AuctionItem
  - pk (default primary key)
  - property: is_expired (BooleanField)
  - product_name (CharField)
  - seller (Foreign key:user)
  - product_value (FloatField, min_val = 0)
  - starting_price (FloatField, min_val = 0)
  - min_bid_increment (FloatField, min_val = 1)
  - final_price (FloatField)
  - start_time (DateTimeField)
  - end_time (DateTimeField)
  - product_image (Filefield)
  - content_type (CharField)
  - product_description (CharField)
  - bid_count (Integerfield)
  - property: highest_bid
  - property: time_remaining (DateTimeField)

- Bid
  - pk (default primary key)
  - auction_item (Foreign key: AuctionItem)
  - bidder (Foreign key: User)
  - time (DateTimeField)
  - price (FloatField)

Backlog

- Auction price real-time update (websocket) - next sprint
- Finalized data models - next sprint
- Finish frontend UI implementation - next sprint
- MySQL database
- Deployment
- Search (optional)
- OAuth (optional)

Thanks for listening!