Best Bids

By-
Dhruv Gupta (17074006)
Dhruva Mahajan (17074007)

Index

Introduction

Best Bids is a bidding site that facilitates consumer to consumer sales through website. Every consumer can put up his product for auction and can bid on products listed by other users.

Auction Products

These are products in ongoing auction.

Ongoing
Bought
Sold
My Product
Register Product

Picasso painting

19 century painting

go to link

Gogh painting

colour heaven

go to link

luxurious home

2000 sq ft

go to link

What our project is able to do.

This project is Django based application.

User can signup and then login very easily to use the site.

User can view all the products he has put on auction and can also see all the items he has bought through auction.

He can also see the list products he has sold and has freedom to list items for bidding.

User with admin control has power to delete user, products.

Requirements

- Django==2.0.6
- django-crispy-forms==1.7.2
- django-filter==2.0.0
- djangorestframework==3.8.2
- Pillow==5.2.0

Database design

1. Relational Schema

2. We have majorly used two tables in our project and has implemented it using models.

3. The tables are

   A. user.

   B. productpage_product.

1.USER

Attributes are-

- **Id** - This is a unique number for every user.
- **Password**
- **is_superuser**- if yes it gives the user administrative control to add or delete products.
- **Username**
- **First_name**
- **Email**
- **last_name**

User

- Id- integer NOT NULL PRIMARY KEY AUTOINCREMENT
- Password- varchar (128) NOT NULL
- is_superuser- bool NOT NULL
- Username- varchar (150) NOT NULL UNIQUE
- First_name- varchar (150) NOT NULL
- Last_name- varchar (150) NOT NULL
- Email- varchar (254) NOT NULL

2. PRODUCTPAGE_PRODUCT

Attributes used are-

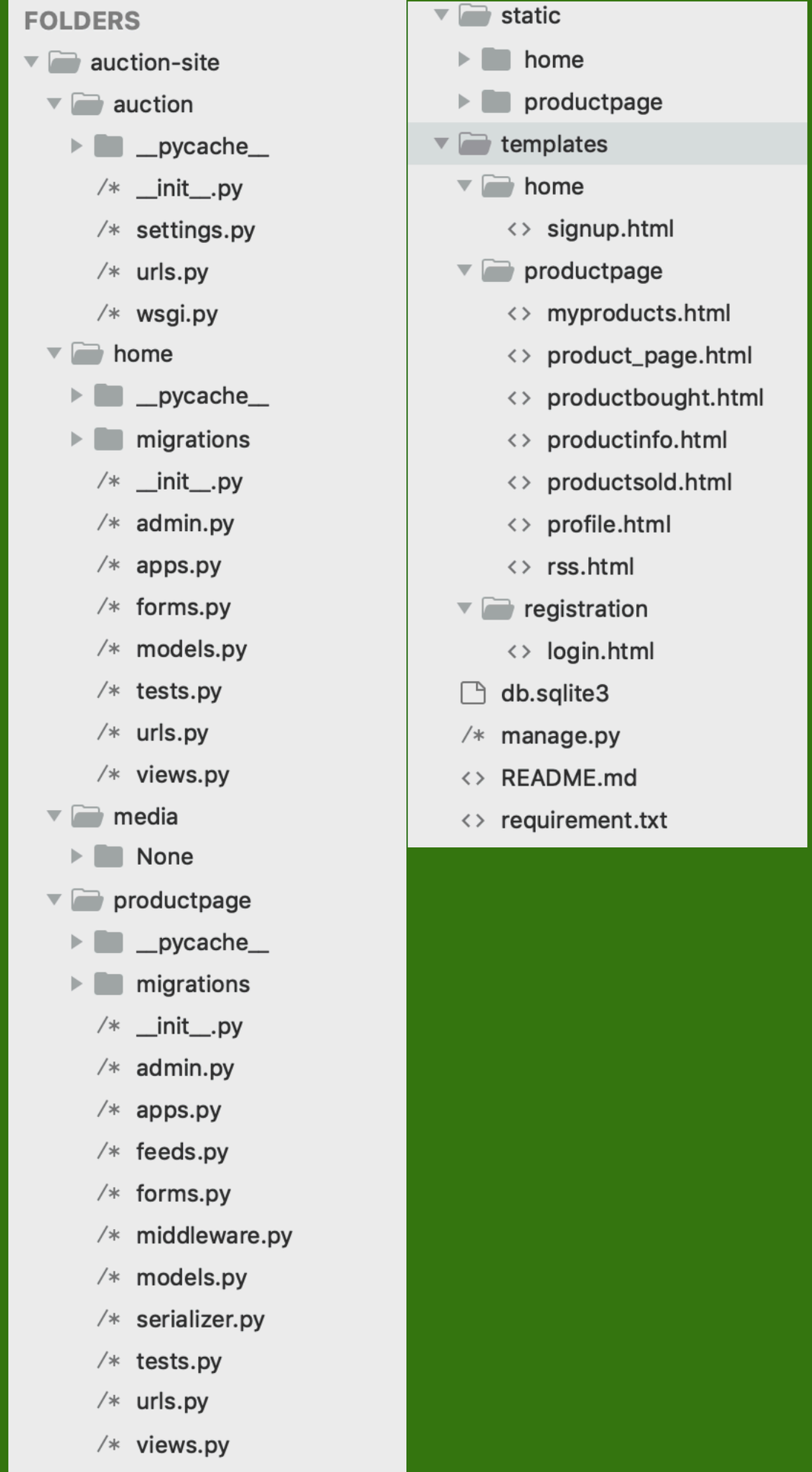
- **Id** - unique number for every product registered on site.
- **Itemname**.
- **Description**.
- **Image**.
- **Initial bid**.
- **Created date**.
- **Status** - Whether the product is on auction is already sold.
- **Buyer**.
- **Bid**.
- **Owner id**- The id of user. Same as id in table user.

productpage_product

- Id- integer NOT NULL PRIMARY KEY AUTOINCREMENT
- Itemname- varchar (200) NOT NULL
- Description- varchar (200) NOT NULL
- Image- varchar (200) NOT NULL
- Initial bid- integer NOT NULL
- Created date- datetime NOT NULL
- Status- integer NOT NULL
- Buyer- varchar (200) NOT NULL
- Bid- integer NOT NULL
- owner id- integer NOT NULL

Project structure

- The project name is auction.
- All the images of the products are stored in media folder.
- There are two apps in the project.
  - Home.
  - Product page.



Functionalities

- One of the app is **home**. This app just handles the login and the signup pages. It is used to render the html pages of login and signup along with css.

- The other app is **product page**. This is the main app which handles all the backend processes of the site.

Productpage app

Functions in views.py

- profile- This function renders the html page “productprofile/profile.html”.
- products- This function renders the html page “product_page.html” which displays all the products which are currently in auction.
- getmyproduct- This function renders the html page “myproducts.html” which displays all the products that are registered by the user currently logged in.
- productsold- This function renders the html page “productsold.html” which displays all the products sold by the user currently logged in.
- productbought- This function renders the html page “productbought” which display all the products bought by the user currently logged in.

Productpage app

Functions in views.py

- model_from_upload- This function helps the user to upload the product.
- newsfeed- used to render html page “rss.html” which displays the rss feed.

Mapping in productpage app.

- ‘ ‘- calls products function in views which displays all the product listed for auction.
- ‘myproducts’- calls getmyproduct function in views which displays products that are listed by the current user.
- ‘productsold’- calls productssold function in views which displays products that are sold by the current user.
- ‘productsbought’- calls productbought function in views.py which displays products that are bought by the current user.
- ‘uploadproduct’- calls the function model_from_upload in views.py which allows the user to register the product on the site.
- ‘newsfeed’- calls the function newsfeed which displays RSS feed.

Use of middleware in the project.

The custom middleware used in the project is a very simple middleware which on request display the message “middleware is working” in the terminal. And on response it displays the message “it’s working again” in the terminal.

Use of syndication in the project.

- Syndication in the projects used in the form of RSS feed which updates user about the science news from different websites using google news.

Addition technologies used in the project.

- API's has been used for fonts and colours of text and other UI.

- Bootstrap is also used.

References

- Django Girls tutorial.
- Tango with Django book.
- Django docs.
- Python docs.
- Django-rest framework.

We are open for suggestions for the website.