Our Project

Goal: To learn how to leverage the Diango framework to build a reassuringly secure and scalable e-commerce website.

Objective: To use application development techniques learned at the University of the Pacific to create an auction-style website using the Django web framework that utilizes all the key coursework we have learned including data structures. algorithms, database management, web development, interface design, computer networking, and parallel computing.

Implementation

Environment

Web framework:

Diango Database:

SQlite3 **Development Tools:**

Virtual Environment Linux: Fedora/Ubuntu

Hosting:

Debugtoolbar Raspberry Pi 4

Languages

Front End:

iQuery/AJAX JavaScript HTML/CSS

Backend End:

Python SQLite REST

Internet Access:

Ngrok



Auction House

Key Features

User Account Management

- · Password Hashing
- · Account Recovery System
- · AJAX Field Verifier · Track Login Attempts
- · Address Management Account Preferences
- · Banning Accounts

PayPal Integration

- Order & Transaction Creation
- · Integrated PayPal Diango API · Instant Payment Notification
- · Sandbox Accounts
- Order Confirmations
- · Paypal Signal Handling (Ngrok)

Pay Pa

Auction **Functionality**

- Various Auction Types
- Fuzzy search Products
- AJAX Auto Updating Bids
- · Bid Management & Settings
- · Buy It Now Functionality **Bid Sniping Prevention** Back End Auction Threading
- · Image Storage System

Purchases & Orders

- · Cart & Checkout
- Transactions & Receipts
- · Purchase & Order History
- · Bid History
- Shipment Tracking **Email Notifications** Reputation/Review System
- Help Center

Evaluation

Efficiency

Query Reduction

- · Prefetching Related
- · Selecting Related

Security

- Password Hashing **SQL** Injection Protection
- Cross Site Request Forgery
- UFW Firewall Limiting Login Attempts
- · Session security

Conclusion

Conclusion: Over the course of the semester we were able to integrate all of the desired features into our application. We were able to utilize the Django framework, demonstrating our expertise in: data structures, algorithms, database management, web development, interface design, computer networking, and parallel computing.

Out of Project Scope:

- Additional Payment Methods
- Shipping integration with USPS
- jQuery load more content on page'

Martin Zacarias - Jason Van Bladel - Colton Remmert

Implementation











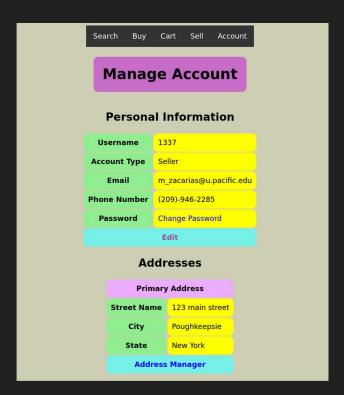








Light Mode



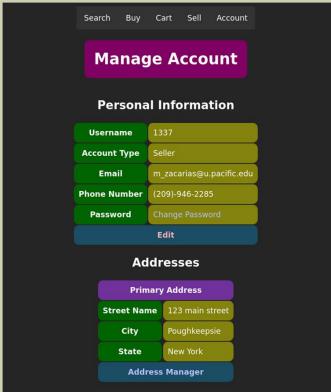
Dark Mode

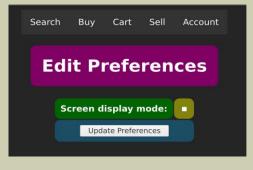


User Management











Login Features

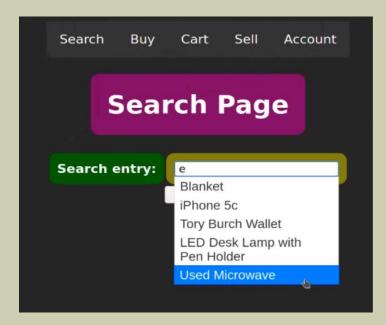


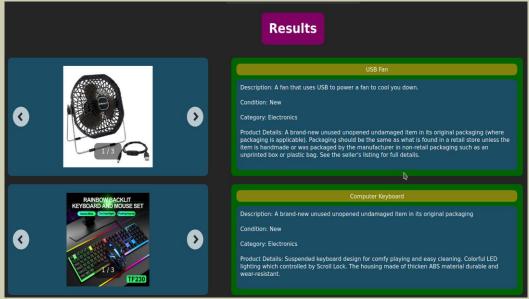




S	seniorprojectauctionhouse@gmail.com <seniorprojectauctionhouse@gmail.com> 7:41 PM</seniorprojectauctionhouse@gmail.com>	
To: c_remmert@u.pacific.edu		
Hello user1,		
Your code is: EV*f3ExJ		
If you did not request to change your password, please contact us and reset your password.		
Thank you,		
Auction	House Website Team	

Searching Auctions

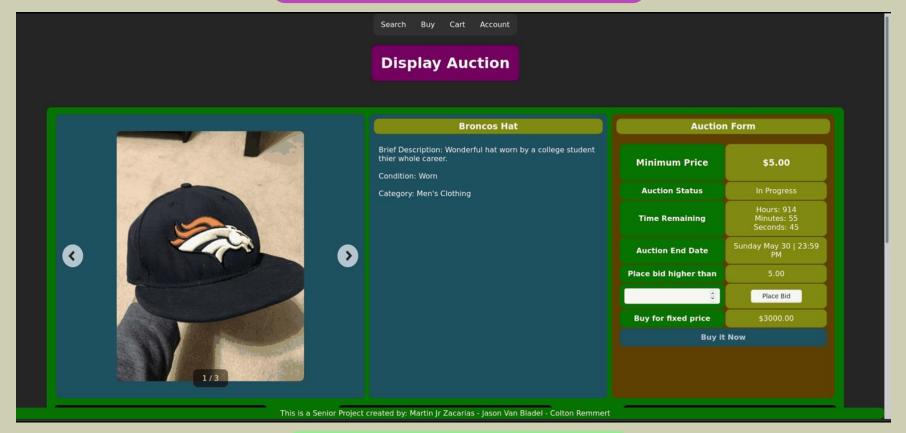




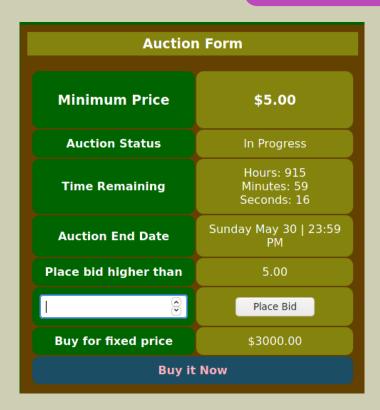
Suggestive Search

Search Result For: Electronics

View Auction



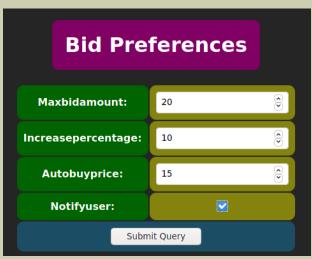
Bid Placement





Bid Management







Anonymous Auction

Auction Details			
Auction Type	Anonymous Auction		
Rules	Current highest bid is not shown		
Minimum bid price	5.00		
Start date	Tuesday Mar 23 10:00 AM		
End date	Sunday May 30 23:59 PM		
Shipping	\$9.00		



Auction Countdown

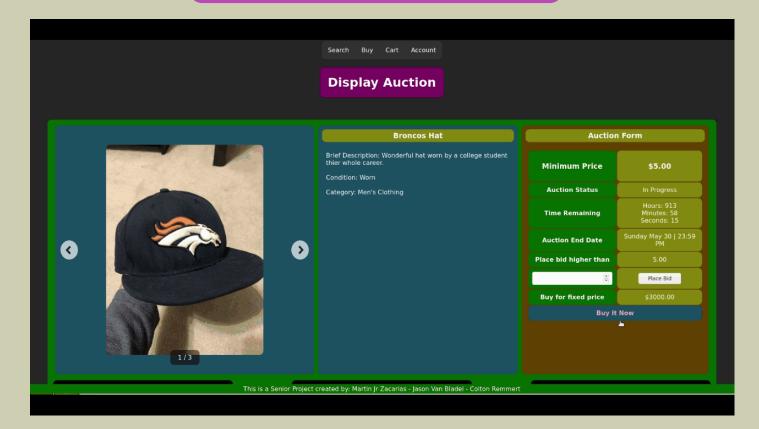
In Progress Auction Status Hours: 0 Time Remaining Minutes: 0 Seconds: 5 Friday Apr 23 | 15:49 Auction End Date Place bid higher than 8 50

Cart & Checkout

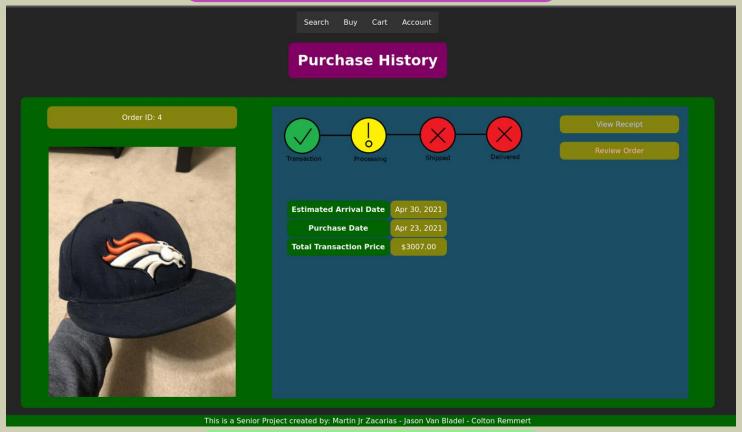




PayPal Integration

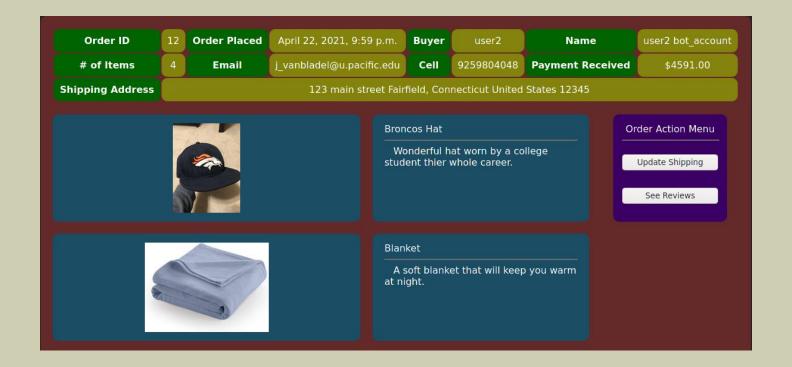


Purchase History

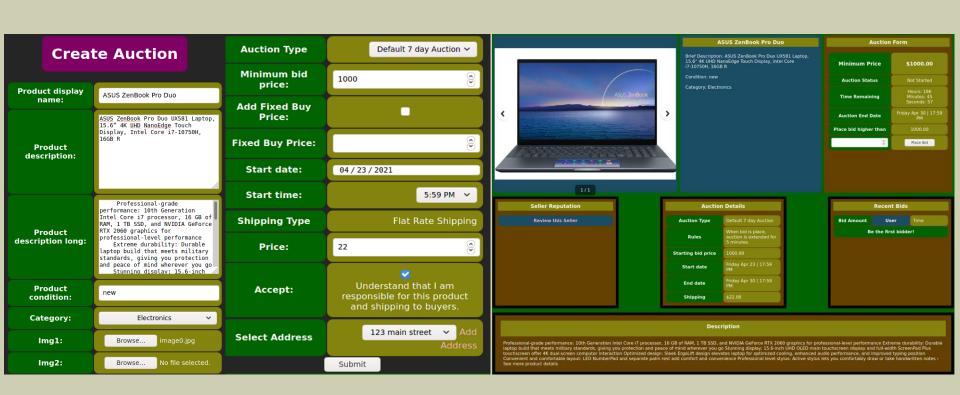


Martin Zacarias - Jason Van Bladel - Colton Remmert

Order Manager

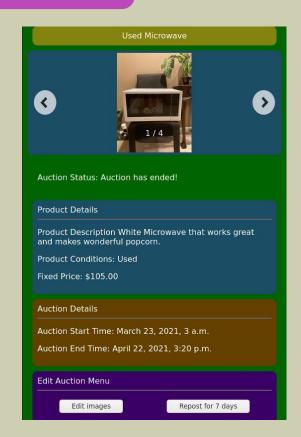


Auction Creation



Auction Manager



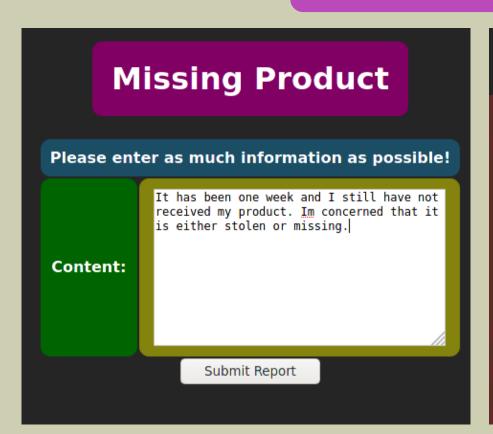


Review System



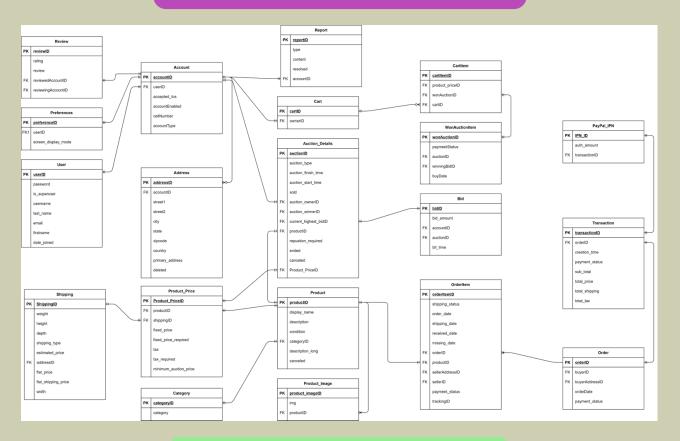


Help Center



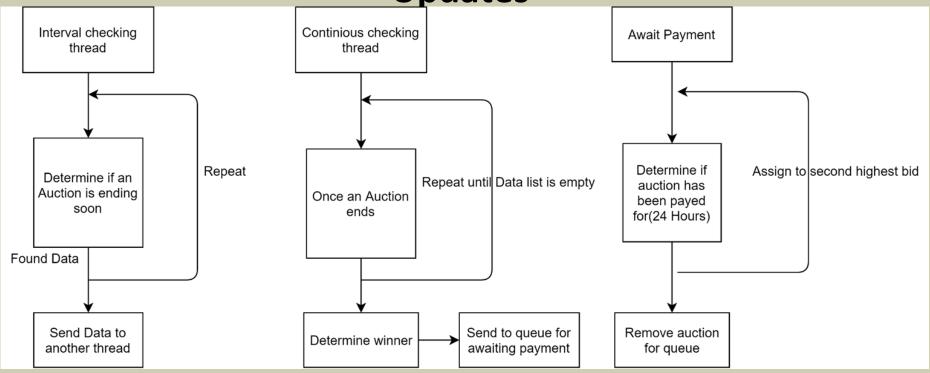


Database Schema



Live Server

Updates



Security Features

id	password
Filter	Filter
4	pbkdf2_sha256\$216000\$cJom6xNhSs1u\$XAdKMCcKUxuT9AOs02UxHgHIw81ARr+rQLMuY/aW2Y0=

Try again later...

Sorry, you have attempted too many login attempts... Come back again in 15 minutes.

Password Hashing

Cross Site Request Forgery

Limiting Login Attempts

```
if form.is_valid():
    #loginForm is Valid
    userLogin = form.cleaned_data['login_username']
    userPass = form.cleaned_data['login_password']

#check user with Django authenticate
    user = authenticate(username=userLogin, password=userPass)
```

SQL Injection Protection

Query Reductions

