

Summer 2019: Capstone

Community Lending Library

Capstone: Summary of what will be covered

- Introduction
 - Inspiration for the project
 - Some challenges that I faced
- Review the anatomy of the site
 - Listing Pages and Item Details
 - Search
 - Request / Checkout Funnel
 - Administrative Pages
 - Notification Emails
 - User Registration & Profile
- Upcoming features
- Questions

Capstone: Listing pages and Item Details

Listing pages

- All Items - homepage
- Item Listings by Category
- Item Listings by Owner
- Search Results Listings

Detail page

- Item Details Page (*design in-progress*)
 - Replacement Cost is the only item detail not shown elsewhere

Capstone: Search

Search Fields

- Title: full-word search using regex
- Description: full-word search using regex

No Results Suggestions

- The page is a mockup (hardcoded results for “wagon”)
- The original idea: show local results from Craigslist
- Other idea: Sponsored search results (Next Adventure for Camping)

Capstone: Request / Checkout funnel

- **Request an item** (homepage, search, category, owner pages)
 - Borrower browses or searches for an item and submits a checkout request
- **Approve an item request** (My Requests page)
 - Owner sets a Due Date when approving the item request
- **Deny an item request** (My Requests page)
 - Owner selects a Reason for declining the item request
- **Check In an item** (My Items page)
 - Owner checks item in (Completed Date is set automatically)

Capstone: Administrative pages

- Pending Requests

- Requests from Borrowers for your items

- My Checkouts

- Your requests for and checkouts of other Users' items

- My Items

- Your items (including request/checkout history, Check In button)

- Manage Items

- Add, Edit, and Delete items from the catalog

Capstone: Notification Emails

.....ACTION.....

A user registers

A user updates their information

A user updates their password

A borrower requests an item

An owner declines an item request

An owner approves an item request

An owner checks in an item

An owner adds an item

An owner edits an item

An owner deletes an item

.....RECIPIENT.....

User

User

User

Borrower & Owner

Borrower

Borrower

Borrower

Owner

Owner

Owner

Capstone: User Registration & Profile

Registration

- Leverages Django's built-in User system

User Profile

- Edit first & last name, email
 - Maybe allow edits to username in the future
- Change Password
- Check Existing Password
 - Note: This is a leftover from development and testing

Capstone: Upcoming Features

Location-based results • Registration confirmation step • Asynchronous email delivery • Owner-initiated checkouts • Batch Request/Checkouts • Private share groups (teams, hobbyists, etc) • Integration with Craigslist • Partnership or sponsorship • Moderation/moderators • Ratings for owners & borrowers • Allow multiple images per item • Resize large images • Add more categories • Reporting per item or owner • Late fees & online payments • Due Date extension or unlimited • Reservations for an item • Message board or way to request items not in catalog

Improve mobile and tablet experience • Use Materialize, Bootstrap, or Vue.js • Switch to PostgreSQL or MySQL

Capstone: Questions?

```
import random
import re

def answer(q):
    ynm = ["Yes!", "No!", "Maybe!"]
    qa = {"who": "The identity is unknown.",
          "what": "Cannot predict now.",
          "when": "When the time is right.",
          "where": "The path is unclear.",
          "how": "There are many ways.",
          "why": "Your question is deep.",
          "are": ynm[random.randint(0,2)],
          "will": ynm[random.randint(0,2)],
          "can": ynm[random.randint(0,2)]}
    character_regex = r'^(w+)'
    question_lower = q.lower()
    start_word = ''.join(re.findall(character_regex, question_lower))
    return qa.get(start_word, "Concentrate and ask again")

def get_all_answers(questions):
    for question in questions:
        response = answer(question)
        print(f"{question}? {response}")

questions = []
while True:
    q = input("Submit your question, or type (done): ")
    if q == 'done':
        get_all_answers(questions)
        break
    else:
        questions.append(q)
```