Book Batch Views

Friday, December 30, 2022 9:31 AM

Home Page:
- Navbar
  - Sign up
  - Log in
  - User search
  - Book search

Users:

Not logged in or not friend profile:
- Display name
- Email
- Reading goal?
- Top quotes
- Public book lists
- Public posts

Friend profile:
- All lists
- All posts
- Activity log
  - Adding to list/making post
  - Can comment on
- Direct message
- Batch membership list
- Start new batch with

Own profile:
- Change name/email/password
- Edit lists
- Make new Post
- Profile picture
- Favorites
  - Genres
  - Books
  - authors

Books:

Not logged in:
- Title
- Author
- Genre
- Quote search

Logged in:
- Add to (list)
- Recommend to (friend)
- Save quote to profile

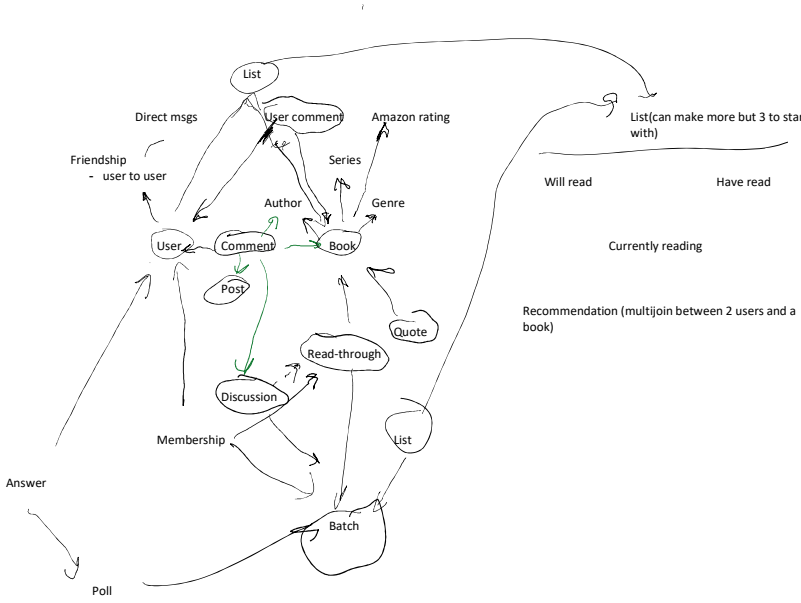
Lists:
- To read
- Have read
- Currently reading

Batches:

Not logged in:
- Batch name
- Currently reading
  - Schedule
- Have read
- Member list

Logged in:
- Polls
- Discussions

Public Blog Posts (works like any forum but designed to focus on original post):
- Can create or comment on created if logged in
- Creator
- Topic



Base template:

Navbar
- Home page
- Sign up
- Log in/{Your username}

Search bar
- Book search
  - Browse top lists
    - Button link
    - Public only
- Batch search
  - Browse top batches
    - Button link
    - Public only
- User search

Footbar
- Log out
- Flash messages

Libby advertising page

Signup:
- username
  - Unique
- Password
- Confirm pw
- Gmail (optional)
- Bio (optional)

Login:
- Username
- Password
- Forgot pw
- Sign up

Emoji Reactions?

Post(make/comment/edit)

Comment(make/view)

Book
- Title
- Author
- Genre
- Published by, series, etc.
- ISBN
- Amazon rating
- Page data:
  - Comments
  - Highlighted quotes
  - Batches reading/have-read/will-read
  - Add to:
    - Batch queue (Batch User reviewal)
    - Wish list
    - Start read
    - Mark read
    - Recommend to: (dropdown)
  - Buy

User
- Username (add friend/edit/message)
- Currently reading
- Email
- Bio
  - User stats (see below)
- Lists
  - Wish list(will read) (can't delete)
  - Have read (can't delete)
- Batches

- User stats
  - Finished reading ___ @__
  - Currently reading ___ since @__
  - Joined ___ batch @__
  - Friends with __ since @__

Batch

Discussion

Messages(conversation)

Models:

Book Search
- Search bar
- Search btn
- Results list

User search
- Search bar
- Search btn
- Results list

Friendship
- user_id1 PFK?
- user_id2 PFK?
- create_date

wish:
- book_id PFK?
- user_id PFK?
  - Delete all wishes upon user deletion
- create_date

user
- id
- display_name
  - unique
- password
  - 8 letter minimum
- email
  - optional(url)
- current_book
  - book_id FK

post
- id
- title
- author
  - user_id FK
- create_date
- content
- topic(book/genre/custom)
  - book_id/genre/null

post_comment
- id
- post_id FK
- author
  - user_id FK
- create_date
- content

- create_date

Books saved to db on addition to any action taken "beyond" the search page (eg. Recommend to friend)

book
- ISBN (PK)
- title
- author
- genre
- record_create_date

quote
- id
- book_id FK
- location_info
- content
- saving_user
  - user_id FK
- create_date

Hosting?
- Let's google, could affect architecture
- pip install gunicorn

Process
- Make temporary database
- Implement basic user authentication flask app
  - app.config[]
  - Save a copy for later
  - Login, signup views
    - Extends base view
    - Nav-bar link buttons
  - User model
- API request
  - Book model
  - Quote search/model
- Search page
  - async javascript search on dedicated page for all books in books table
  - Search button makes API search request
    - List model
- Book page
- Quote search
- User page

Alright. So. Information flow.

Data comes from API and is handled by…. Server? Server is slow. Client? How do I save the data from the client into the database? Through the flask app. So javascript takes the data using axios, and the book data is handled by the client entirely. Loaded into rows with the ISBN that can be passed back into the API with the model-matching data.

SO

Search => book data (async javascript loading) -> save =>
flask-app
- invoked to manipulate database. Pulls data from database for all information, javascript handles all API communication

Process again
- Book search page
  - Adjust book model
  - Save book
- User authentication
  - Basic model made already
  - Make sure authentication works
  - Implement basic authorization
  - So I can add time/user of add from search
- Break

Templates:
Home
Users/
- Signup
- Login
- Search
- User_id

Books/
- Search
- Book_id

Anything I want to display to do with the database information has to be stored directly in the models, doing math on it in javascript will have to be taken from the queryselector.value (document load-in values)