HofTAPS

By

Awab Abedin, Robbie Bernstein, Thomas Catania, Sanjit Menon,
James Nastasi, Marcus Wiesenhart

The Problem

Price:

- Textbooks are expensive.
- Only used for one semester and collects dust after.
- Not worth the price relative to the amount of use.

Accessibility:

- Exact type of required textbook may be out of stock.
- Finding a textbook that is out of date may become unreasonable.
- Not having access to the textbook can impede course progress.

Other Solutions/Background

Amazon

- Popular solution for buying and selling textbooks.
- Can buy books new and used.
- Pricing can be inconsistent; new books are expensive.

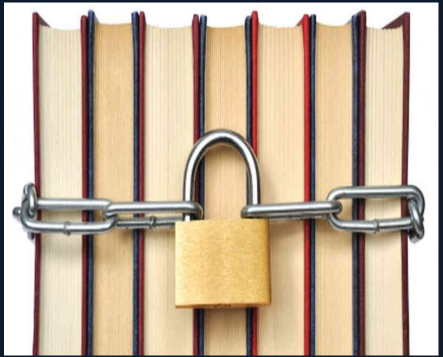
eBay

- Popular site for purchasing used textbooks.
- Prices are often fair; Bidding options and flat prices are available.
- Listings are inconsistent; they may not have the required textbook.

Our Solutions

A Platform Students Can Rely On

- A haven for students to buy books at a reasonable price
- Competitive marketplace - Secure the bag!
- Only textbooks get exchanged, not passwords or data

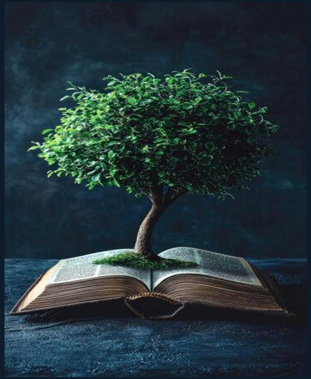


Designed to save time and reduce effort

- Minimalist Design
- Quick Access: No complex menus

Sustainability

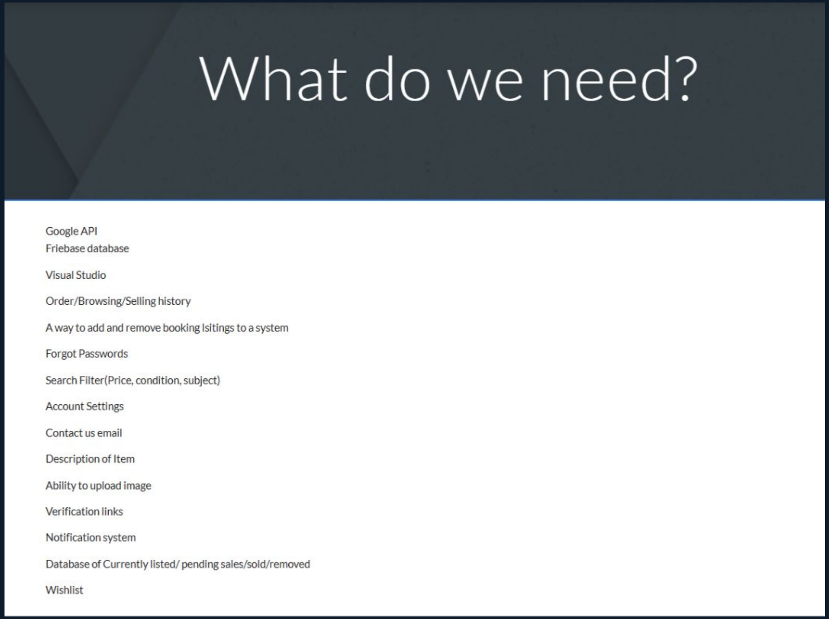
- Save the planet - one overpriced textbook at a time!



Project Goals

Develop intuitive textbook exchange platform

- Have a easy to understand User Interface
- Enable easy searches with sorting and filtering
- Facilitate fast and efficient communication between buyers and sellers
- Have a record of previous transactions
- Use the Google Book API To pull information about the book
- Have textbooks/listings, users, and notifications stored on a database
- Remove textbooks/listings after the a purchase



What do we need?

Google API
Friebase database

Visual Studio

Order/Browsing/Selling history

A way to add and remove booking lsitings to a system

Forgot Passwords

Search Filter(Price, condition, subject)

Account Settings

Contact us email

Description of Item

Ability to upload image

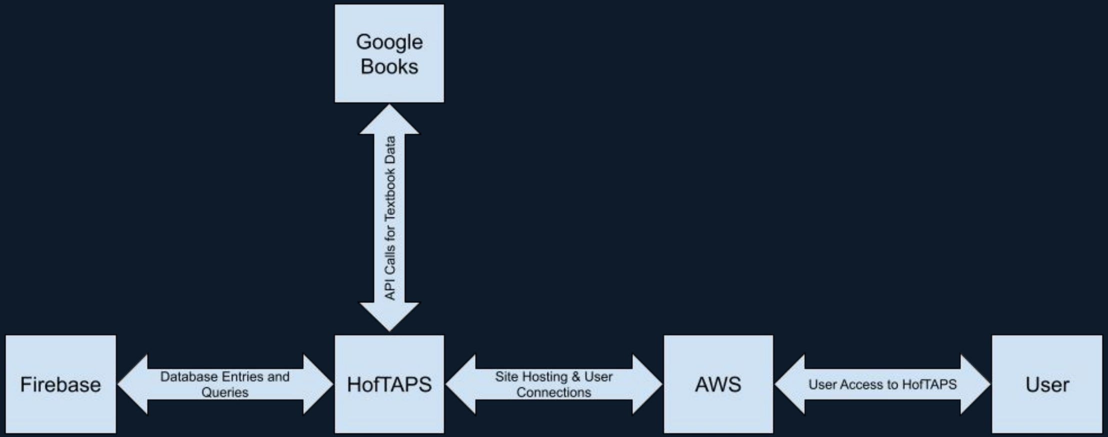
Verification links

Notification system

Database of Currently listed/ pending sales/sold/removed

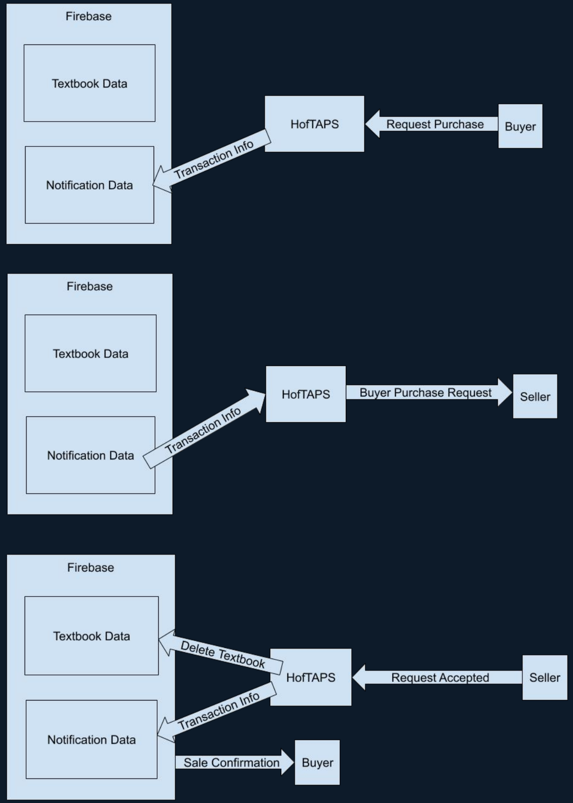
Wishlist

Project Design


Google Books
API Calls for Textbook Data
Firebase
Database Entries and Queries
HofTAPS
Site Hosting & User Connections
AWS
User Access to HofTAPS
User

- Main Components: HofTAPS Source, Firebase, Google Books, AWS, User
- All communication between user, database, and API are handled through our application
- Tools include Firebase, Google Books, and AWS

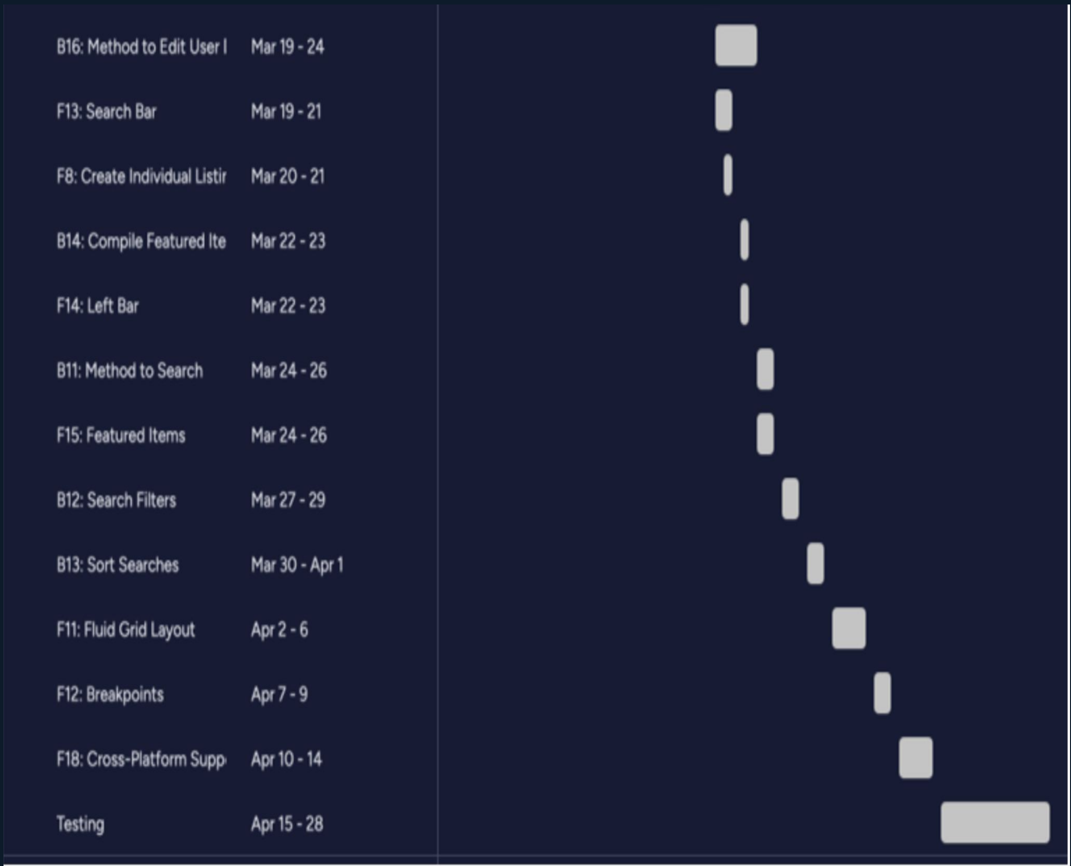
Methods



- User requests book, seller accepts, textbook removed
- Requests are sent over in-site notifications
- Sellers accept/decline sales over notifications
- Once a seller accepts, the textbook is removed from Firebase

Implementation

- Project Phases: Database Integration, Authentication, API Integration, Account/Information, Listings, Notifications, Testing
- After each phase completion we would integrate, debug, improve, and then update the product
- Issues Faced: Firebase data sync bugs during testing, Version control merge conflicts, Time management with class deadlines



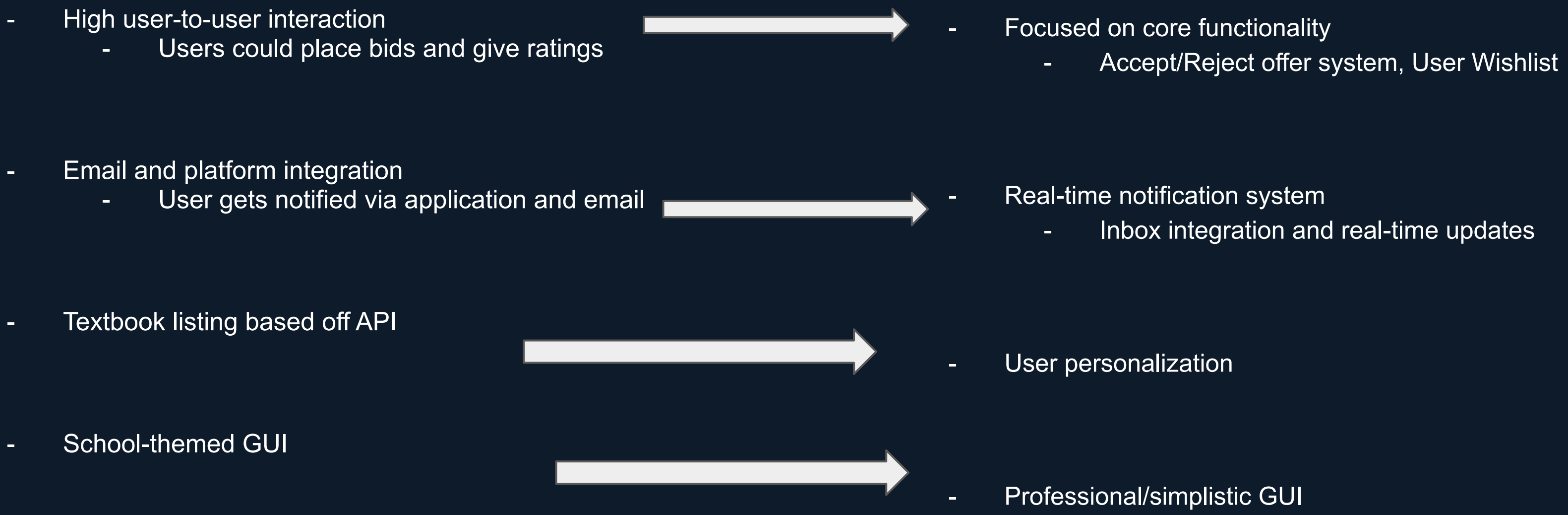
B16: Method to Edit User I	Mar 19 - 24
F13: Search Bar	Mar 19 - 21
F8: Create Individual Listir	Mar 20 - 21
B14: Compile Featured Ite	Mar 22 - 23
F14: Left Bar	Mar 22 - 23
B11: Method to Search	Mar 24 - 26
F15: Featured Items	Mar 24 - 26
B12: Search Filters	Mar 27 - 29
B13: Sort Searches	Mar 30 - Apr 1
F11: Fluid Grid Layout	Apr 2 - 6
F12: Breakpoints	Apr 7 - 9
F18: Cross-Platform Supp	Apr 10 - 14
Testing	Apr 15 - 28

Testing

- Testing Strategy: Manual Testing
- Testing Methods: Regression Testing, Integration testing (important), Black-box/White-box, Unit Testing
- Testing data: Dummy user's, Dummy textbook listings, etc
- Test Results: Cores are functional and flow well; Easy to use interface (minimal bugs)

We preserved old versions of HofTAPS/Utilized Github Branches in case of testing failure

Risks and Scope Changes



Original Scope

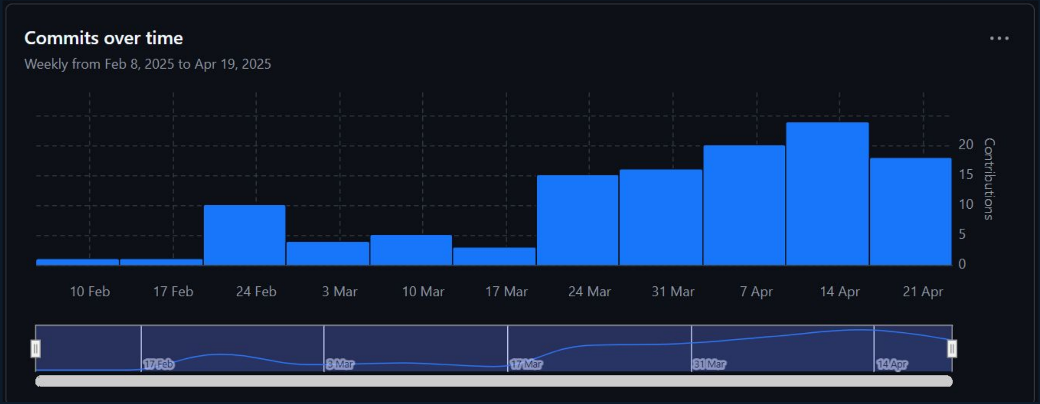
- High user-to-user interaction
  - Users could place bids and give ratings
- Email and platform integration
  - User gets notified via application and email
- Textbook listing based off API
- School-themed GUI

How Scope Changed

- Focused on core functionality
  - Accept/Reject offer system, User Wishlist
- Real-time notification system
  - Inbox integration and real-time updates
- User personalization
- Professional/simplistic GUI

What We Did

- Robbie: Database & Integration
- Thomas: Authentication & Retrieving/Manipulating User Data
- Sanjit: Notifications & API
- James: Front End & Styling
- Awab: Faq Page
- Marcus: Testing

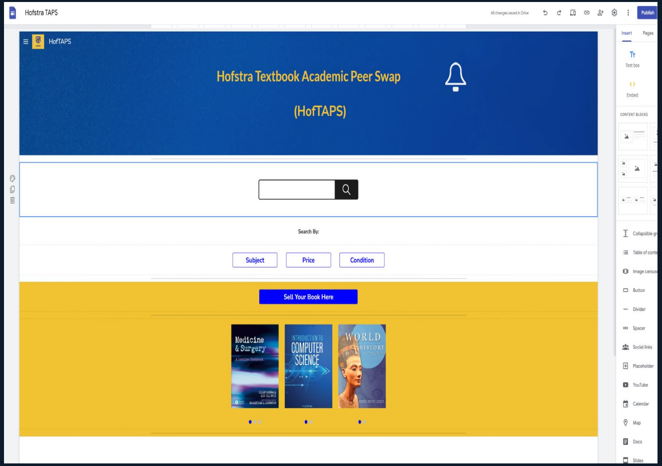


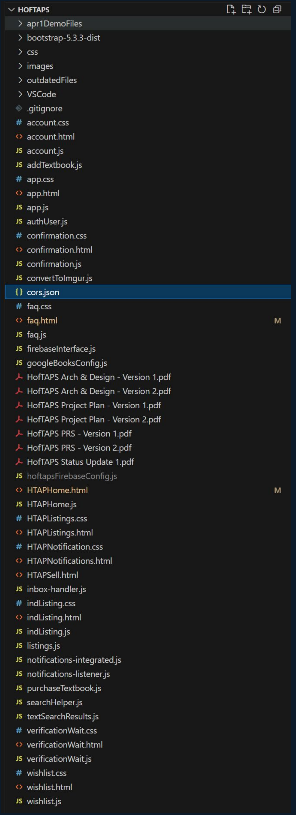
Commits over time

Weekly from Feb 8, 2025 to Apr 19, 2025

Our Team Experience

- Meeting multiple times a week.
- Reviewed the week’s code every Tuesday and Thursday
- We used GitHub , VSCode, Firebase, AWS, Google Book API
- Languages we used: HTML, CSS, JavaScript
- We had 9 Versions.
- Clear vision from the beginning





```
HOFTAPS
  > apr1DemoFiles
  > bootstrap-5.3.3-dist
  > css
  > images
  > outdatedFiles
  > VSCode
  .gitignore
  account.css
  account.html
  account.js
  addTextbook.js
  app.css
  app.html
  app.js
  authUser.js
  confirmation.css
  confirmation.html
  confirmation.js
  convertToImgur.js
  cors.json
  faq.css
  faq.html                                        M
  faq.js
  firebaseInterface.js
  googleBooksConfig.js
  HofTAPS Arch & Design - Version 1.pdf
  HofTAPS Arch & Design - Version 2.pdf
  HofTAPS Project Plan - Version 1.pdf
  HofTAPS Project Plan - Version 2.pdf
  HofTAPS PRS - Version 1.pdf
  HofTAPS PRS - Version 2.pdf
  HofTAPS Status Update 1.pdf
  hoftapsFirebaseConfig.js
  HTAPHome.html                                   M
  HTAPHome.js
  HTAPListings.css
  HTAPListings.html
  HTAPNotification.css
  HTAPNotifications.html
  HTAPSell.html
  inbox-handler.js
  indListing.css
  indListing.html
  indListing.js
  listings.js
  notifications-integrated.js
  notifications-listener.js
  purchaseTextbook.js
  searchHelper.js
  textSearchResults.js
  verificationWait.css
  verificationWait.html
  verificationWait.js
  wishlist.css
  wishlist.html
  wishlist.js
```

Conclusion

- This project went smoothly and with no major issues.
- Learned:
  - How to store and edit data on a database
  - How to connect to an API
  - How to use GitHub
  - Working in a team
  - How to deploy a website
  - Connecting front end and back end
  - Integrating different peoples code