Book Buddy

Sprint 3 Update

Team 5 - Code Avengers

Pace University Capstone Project - Spring 2024

Agenda

1. Team Details
2. Improvements Made
3. Project Description
4. Working Agreement
5. Personas
6. Technologies
7. Algorithms
8. Diagrams
9. Product Backlog
10. Sprint 3 Backlog
11. Metrics
12. Retrospective
13. Sprint 4 Plans
14. Demo

Team Members

Jimmy Karoly
Scrum Master/Developer

Dhanasri Salla
Machine Learning Engineer

Sai Sandeep Mandava
Machine Learning Engineer

Team Members

Praveen Raj Gujju
Tester

Anusha Meka
Developer

Sairam Vinjamoori
Developer

Team Members

Krushal Rama
Developer

Alekhya Bommineni
Developer

Changes due to Professor’s Feedback

- Used correct story point numbers
- Refined Product Backlog and split User Stories
- Changed formats of Product Backlog/User Stories/Acceptance Criteria to make them more readable
- Changed format of metrics slide to make it more readable
- Reformatted Algorithm slide to make that more readable

Slides with changes have this icon: 

Project Description

Project Name:	Book Buddy
Team:	Code Avengers
Project Description:	A recommendation system that will prioritize unknown and less popular authors and allow users to purchase recommendations. For book readers who search for new books to read the recommendation system, Book Buddy, is a website that allows a user to generate recommendations based off of a survey or books they have read unlike GoodReads or other book recommendation systems our application will prioritze newer and unknown authors with different life experiences and views.
Benefit Outcomes:	A reader will have get recommendations that are different than they have gotten before Unknown and less popular authors will be able to gain a new audience Users will have a one stop site to organize their reading habits and purchase books that are recommended to them
Github Link:	https://github.com/htmw/2024S-Code-Avengers/wiki

Team Working Agreement

Team Agreement

- All team members agree to attend all meetings scheduled every week on Wednesday, either before or after class
- If a team member can't attend a meeting then they will let the group know ahead of time, and we can talk about the possibility of rescheduling and if rescheduling is not possible then the absent teammate will agree to whatever is decided by the majority that attends the meeting
- If a team member needs help then they will reach out to the team in advance and not wait until the night before something is due
- All team members should feel open to sharing their opinion and group consensus will decide the direction we will go in, if not opposing opinions are shared then the assumption will be that you are in agreement.
- Team members will stay active in the group chat and pay attention to questions and participate in group discussions about project details
- Work will be divided as evenly as possible and is expected to be complete in a timely manner
- Members agree to respect the time and other obligations of fellow team members, by attending meetings on time and responding in the group chat in a timely manner.
- Work will be done to the best of every members ability

Personas

User Persona

Alex Müller

About

Age: 62 Years

New York, NY

Education: Bachelor's

Gender: Male

Job Title: Retired Bank Manager

Profile

Alex is an avid book reader and always loves looking for new books. Lately he's been underwhelmed by the recommendations he's gotten from the leading book sites, and wants a place that will specifically recommend underrepresented authors.

Goals

- He wants to find new authors
- He doesn't just want the same popularity based recommendations

Reading Habits

- Alex reads daily
- He can usually finish 3-4 books in a month

Pain Points

- Likes GoodReads but would want one place that could give recommendations and he could purchase them
- Gets recommended the same thing repeatedly

User Persona

Jennifer Farmer

About

- **Age: 26 Years**
- **Miami, FL**
- **Education: Bachelor's**
- **Gender: Female**
- **Job Title: Graduate Student**

Reading Habits

- Jennifer reads daily
- Most reading for school she can finish 2 books of her own choosing a month

Profile

Jennifer is a 26 year old graduate student and she is interested in philosophy and gender studies. She reads a lot for school but also likes to find time to read for leisure.

Goals

- She wants to find books in her field of study that are written by less-known authors
- She wants to find one place for new recommendations and purchasing

Pain Points

- Difficulty finding diverse book recommendations outside the mainstream
- Doesn't have a lot of time to devote to finding new books

User Persona

Joseph Harps

About

- Age: 32 Years
- Los Angeles, CA
- Education: Master's
- Gender: Male
- Job Title: Software Developer

Reading Habits

- Joseph tries to read once or twice a week
- He can usually finish 1 book a month

Profile

Joseph is a busy software developer and doesn't have a lot of time to read. He wants to branch out and find new topics to read about from different perspectives. He wants to experience books outside of the normal bestsellers.

Goals

- Since he doesn't have a lot of time he wants something that is easy to use and doesn't require much setup.

Pain Points

- Doesn't know where to start looking
- Good Reads seems to always recommend the same things

Minimum Viable Product

Our MVP will consist of the following functionalities:

- Registration/Login
- Book Recommendations
- Book Collections for User
- Marketplace Functionality

Technologies and Algorithm

Front End Technologies

We will be using the React JS library to create our UI along with some HTML and CSS as well.

For our login/registration we will be using Google Firebase. They allow you to interact with it through an API

Back End Technologies

We are using PyTorch for our ML Algorithm

We will create an API for the algorithm using Python

The backend will be written in Java using the Spring Framework

Our Database will be Postgres

Machine Learning Algorithm

1. Feature Vector Construction:

   Represent each book as a vector of features. Each dimension corresponds to a specific characteristic or keyword associated with the book.
   For textual features, use techniques like TF-IDF to weight the importance of each word or tag.

2. Cosine Similarity Calculation:

   Calculate the cosine of the angle between two book vectors, A and B, using the formula: **Cosine Similarity = (A · B)/‖A‖‖B‖**

   Where:

   A · B is the dot product of vectors A and B
   ‖A‖ and ‖B‖ are the magnitudes (Euclidean norms) of vectors A and B, respectively.

3. Interpretation:

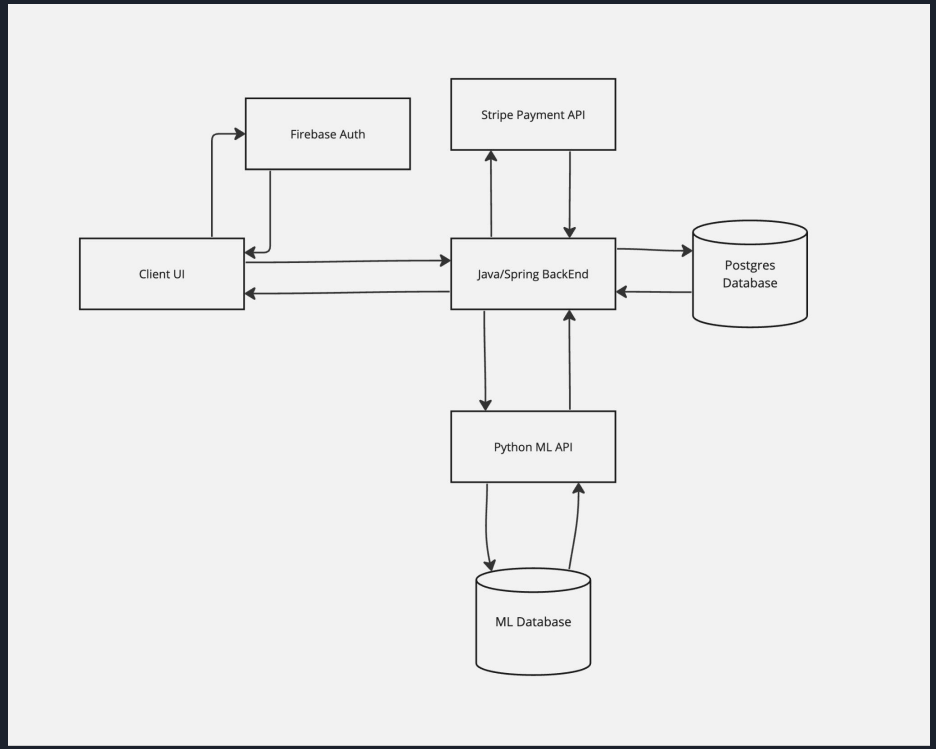
   The cosine similarity ranges from -1 to 1, where 1 means identical, 0 indicates orthogonality (no similarity), and -1 implies opposite.
   For book recommendations, we focus on positive similarity scores, with higher scores indicating greater similarity.

4. Recommendation:

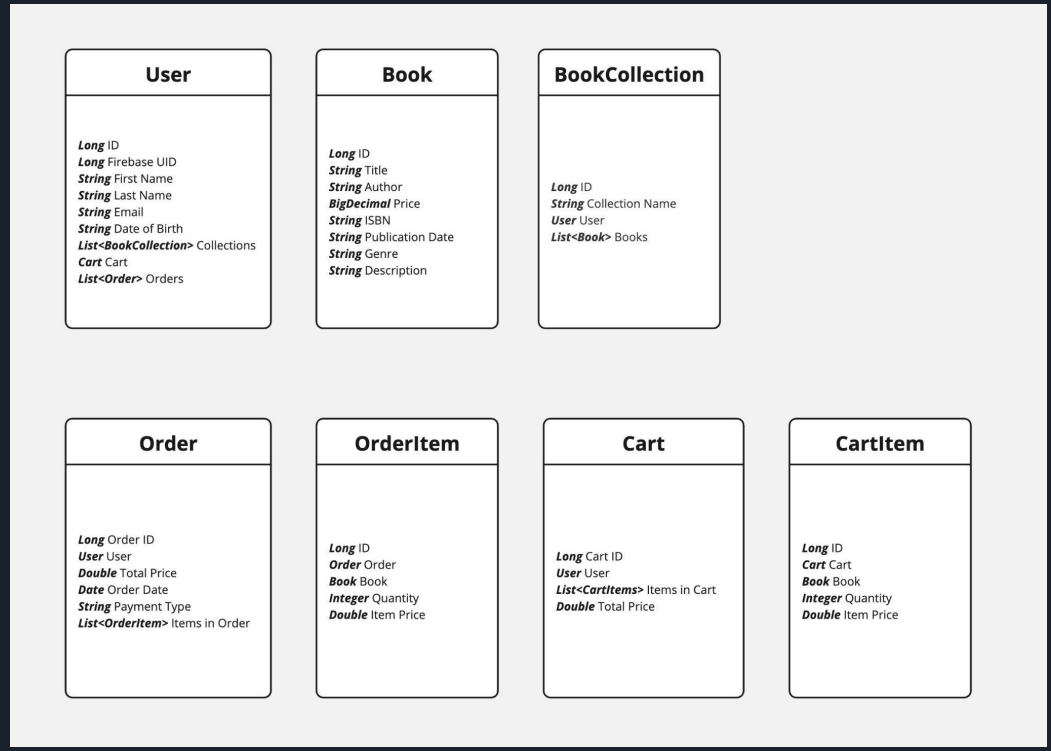
   For a target book, calculate its cosine similarity with all other books in the dataset.
   Recommend books with the highest similarity scores to the reader.

Diagrams

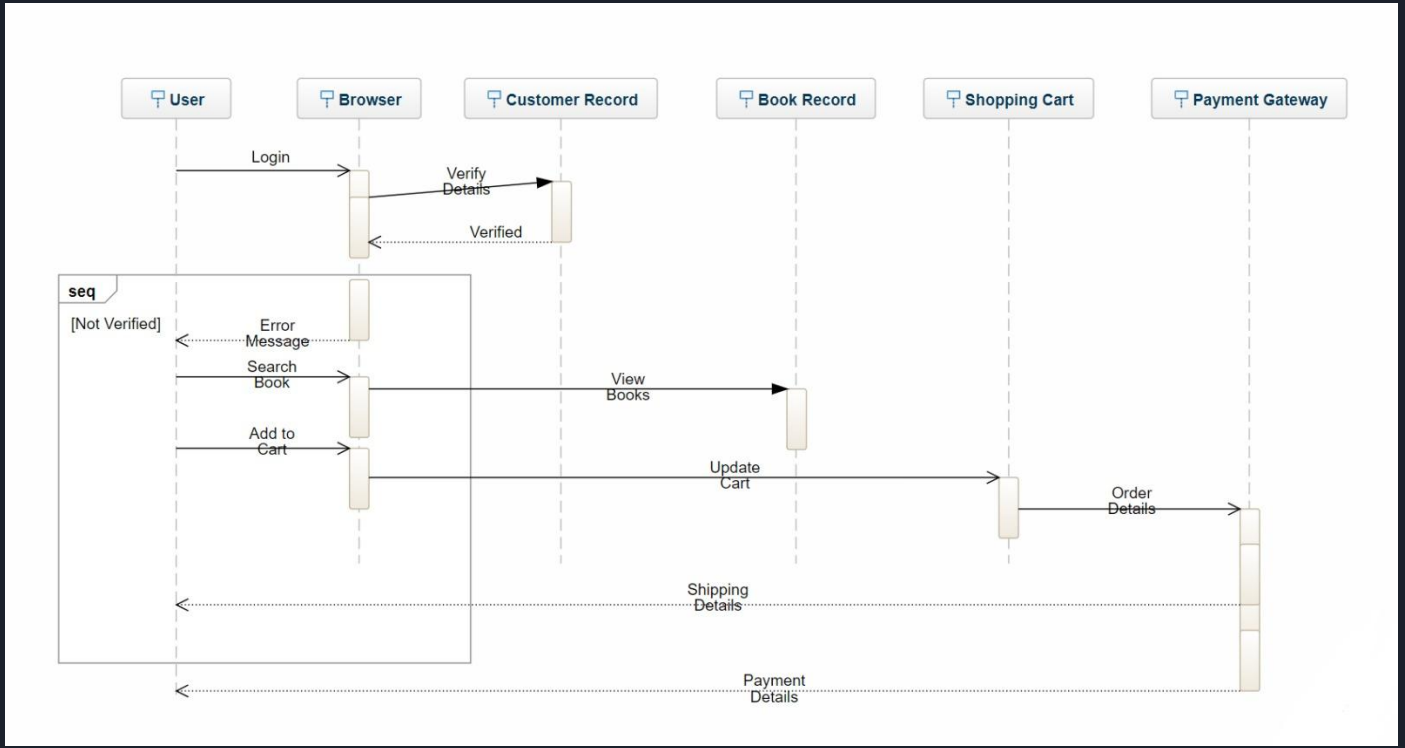
Conceptual Architecture Diagram



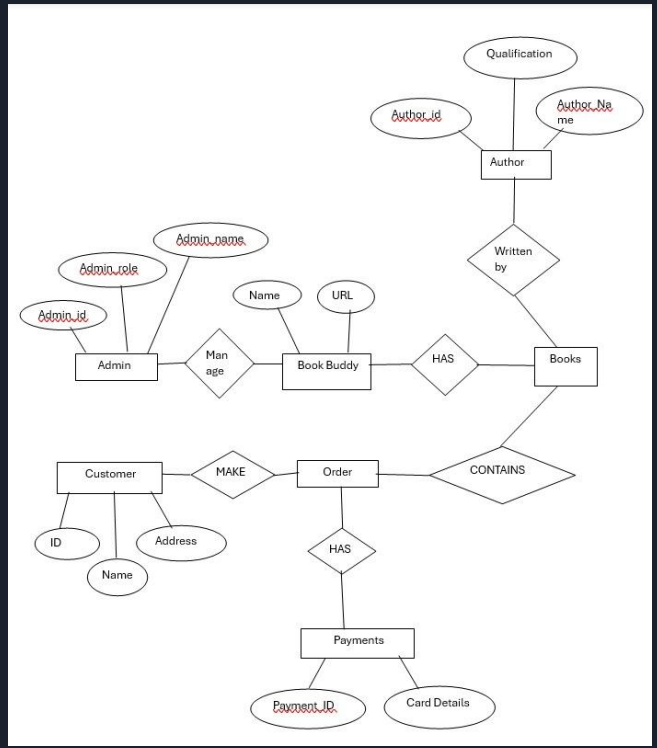
Class Diagram



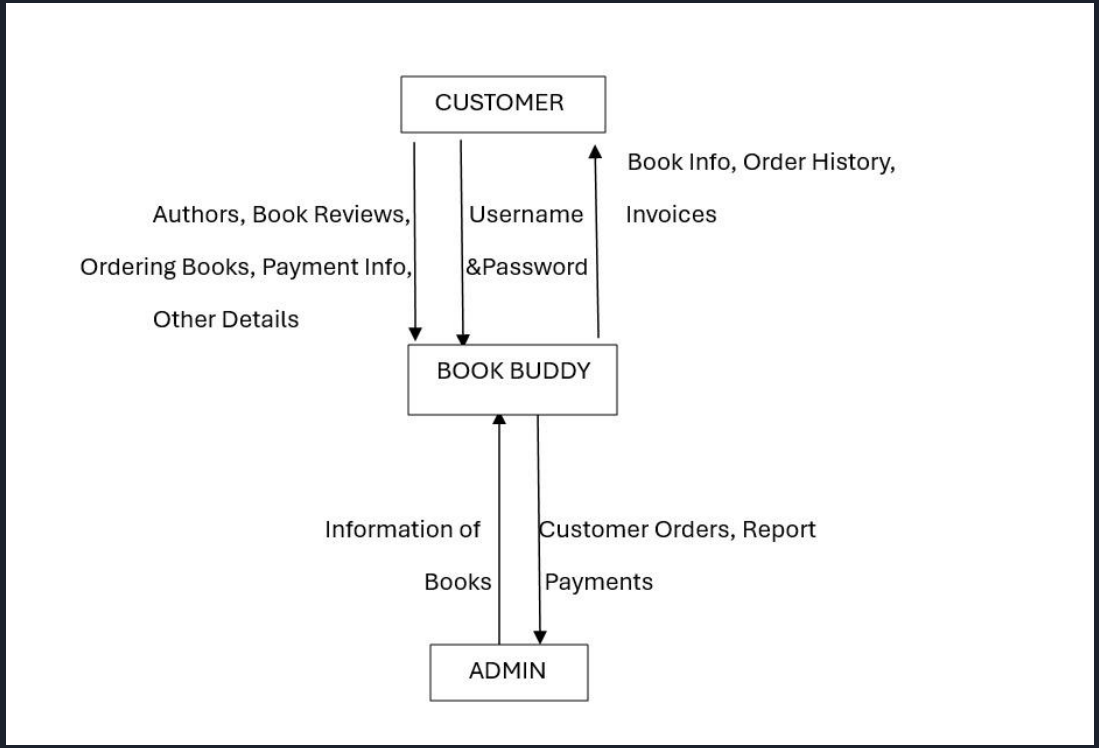
Sequence Diagram



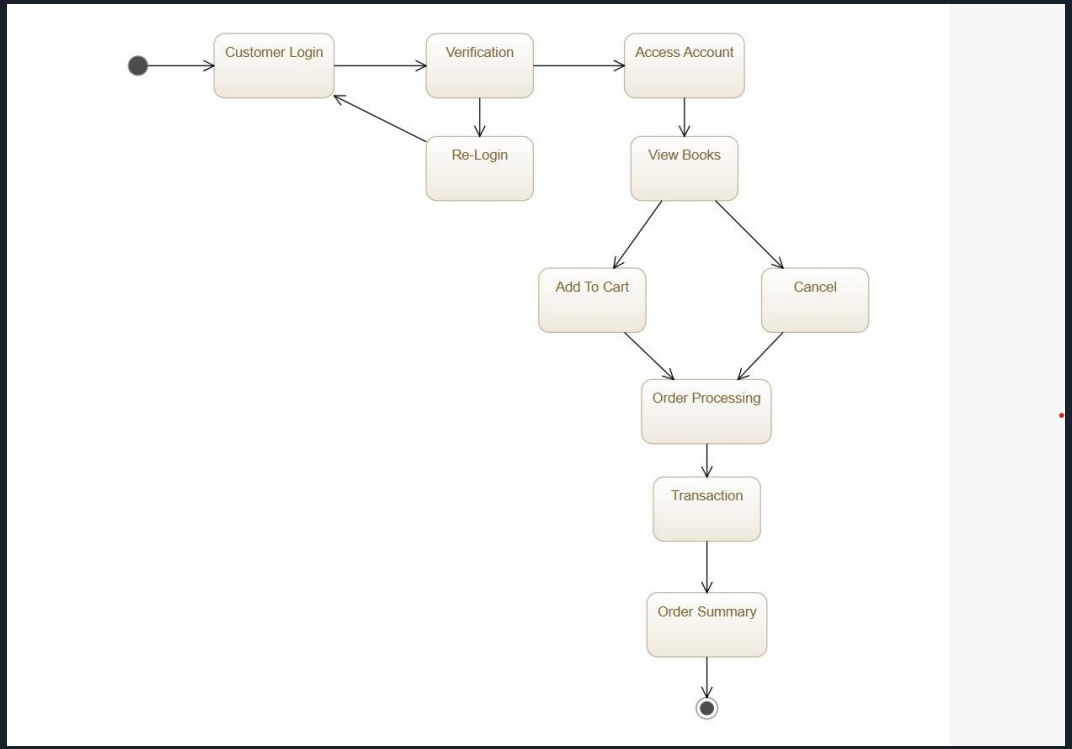
Entity Relationship Diagram



Context Diagram



State Diagram



Sprint 2 Recap

- Found ML dataset and tested algorithm
- Designed main UI and UX for homepage
- Finished experimenting with backend classes
- Finalized project design
- Finalized project architecture
- Completed 27 story points out of 31 total points
- 87% Completed/Committed Ratio

Product Backlog

Product Backlog

ID	User Stories	FEATURE	ESTIMATION
1	As a passionate reader, I want to input my likes and dislikes into the system, so that the recommendation system can take them into account	Recommendation System	2
2	As a passionate reader, I want the system to generate recommendations based on my likes and dislikes so that I can get better recommendations.	Recommendation System	8
3	As a passionate reader, I want to view the recommendations generated based on my likes and dislikes and explore the books from the list.	Recommendation System	2
4	As a passionate reader, I want the system to generate recommendations that promote diversity and highlight authors from different backgrounds	Recommendation System	8
5	As a passionate reader, I want to be able to adjust my likes and dislikes so that I can refine the recommendations.	Recommendation System	3
6	As a new reader with no reading history, I want to receive basic recommendations based on popular and new authors, so that I can read more.	Recommendation System	3
7	As a new reader with no reading history, I want to specify my preferred genre so that recommendations are more tailored to my interests.	Recommendation System	5
8	As a new reader with no reading history, I want to explore recommended books with brief descriptions or summaries to help me decide, so I can read more.	Recommendation System	3
9	As a passionate reader, I want to be able to create a new list that I can name so I can organize and track my reading habits.	Book Collections	2
10	As a passionate reader, I want to add books to a specific list, such as liked/disliked or recommendations, so that I can keep track of books I have read or want to read.	Book Collections	3
11	As a passionate reader, I want to view and manage the books in my lists, including editing and deleting them, so that I can maintain an organized reading list.	Book Collections	5
12	As a passionate reader, I want my recommendations to be automatically added to a specific list for easy access, so that I can keep track of recommended books.	Book Collections	3
13	As a busy reader, I want to view detailed information about a book, including its description, author, and price, so that I can decide whether to purchase it.	Marketplace	5

14	As a busy reader, I want to add books to my shopping cart, so that I can purchase them later.	Marketplace	3
15	As a busy reader, I want to view and manage the items in my shopping cart, including adding, removing, and updating quantities, so that I can finalize my purchase.	Marketplace	3
16	As a busy reader, I want to have a smooth and secure checkout process, including entering my payment and shipping information, so that I can complete my purchase.	Marketplace	5
17	As a busy reader, I want to view my order history, including past purchases and order status, so that I can track my purchases.	Marketplace	8
18	As a busy reader, I want the option to use different payment methods when purchasing books, so that I am not limited to only one.	Marketplace	3
19	As a passionate reader, I want to be able to register for an account with my email address and password, so that I can access the application.	Login/Registration	5
20	As a passionate reader, I want to be able to register for an account with third party providers (Facebook, Google, etc), so that I can access the application and not have to remember another password.	Login/Registration	3
21	As a busy reader, I want to be able to reset my password if I forget it, so that I can regain access to my account.	Login/Registration	2
22	As a busy reader, I want to be able to log out, so I can end my session securely.	Login/Registration	2

Sprint 3 Backlog

Sprint 3 Stories

User Story ID	Sprint 3 Stories	POINTS	STATUS
9	As a passionate reader, I want to be able to create a new list that I can name so I can organize and track my reading habits.	2	Complete
Acceptance Criteria for US 9: The system will allow for the user to input a custom name for the list. The system will persist the name in the database so it can referenced again.			
10	As a passionate reader, I want to add books to a specific list, such as liked/disliked or recommendations, so that I can keep track of books I have read or want to read.	3	Complete
Acceptance Criteria for US 10: The system will allow the user to add a book to a list from the book detail page. The book that is added to the list will be stored in our database. The system will send a confirmation to the front end when the book is successfully added.			
11	As a passionate reader, I want to view and manage the books in my lists, including editing and deleting them, so that I can maintain an organized reading list.	5	Complete
Acceptance Criteria for US 11: The system will implment a way to order books and mantain that order. The user will have control over the order of books. The user will be able to add and remove books from the list.			
12	As a passionate reader, I want my recommendations to be automatically added to a specific list for easy access, so that I can keep track of recommended books.	3	Complete
Acceptance Criteria for US 12: When the ML algorithm sends recommendations, the system will automatically store them in a list titled "Recommendations" with no additional input from the user.			
19	As a passionate reader, I want to be able to register for an account with my email address and password, so that I can access the application.	5	Complete
Acceptance Criteria for US 19: The system will offer registration that is easy to use. The system will walk the user through registration.			
21	As a busy reader, I want to be able to reset my password if I forget it, so that I can regain access to my account.	2	Complete
Acceptance Criteria for US 21: The system will implement a password recovery method. The method should be quick and simple to use.			

22	As a busy reader, I want to be able to log out, so I can end my session securely.	2	Complete
Acceptance Criteria for US 22: The system will have a logout feature. The system will provide feedback to the user that logout is complete. After logout, they should be redirected to the home page. Logging out should clear any session information			
13	As a busy reader, I want to view detailed information about a book, including its description, author, and price, so that I can decide whether to purchase it.	5	Complete
Acceptance Criteria for US 13: The system will populate a detail page for all the books that are stored in our database.			
14	As a busy reader, I want to add books to my shopping cart, so that I can purchase them later.	3	Complete
Acceptance Criteria for US 14: The system will implement a shopping cart system that will persist between sessions.			
1	As a passionate reader, I want to input my likes and dislikes into the system, so that the recommendation system can take them into account	2	In Progress
Acceptance Criteria for US 1: The system should allow for the user to input likes and dislikes.			
6	As a new reader with no reading history, I want to receive basic recommendations based on popular and new authors in genres I like , so that I can read more.	3	In Progress
Acceptance Criteria for US 6: The system should have a way to generate recommendations based off of genres only.			
		35	

Sprint 3 Test Cases

Test Case ID	User Story ID	Title	Objective	Test Data	Steps	Expected Result	Status
TC - 003	US 9	Creating Lists to Organize Books	Make sure the user is able to create a new list in the system, and the API endpoints exist and work correctly	-Stored User in Database -Name for new List	1. Send request to endpoint for creating a new list. Use test data for request. 2. Watch for response from API 3.Double check in database that everything was stored correctly	System will confirm new list was created and database will store correct information connected to the user that created the list.	PASS
TC - 004	US 10	Adding Books to a list	Make sure the user is able to add any book that is in our database to a specific list	-Stored book in database -Created list connected to user	1. Send request to endpoint for adding a book to a list 2.Watch for response from API 3.Check database for correct storage	System will add the book to the specified list. The database will store the correct relations.	PASS
TC - 005	US 11	Manage List (Edit Order and Deletion)	Make sure the user is able to edit the order of the books that are added to the list, also make sure deletion from the list is possible	-Created list with at least 3 books added	1. Send request to endpoint to delete a book from the list. 2. Watch for response from API 3. Check database deleted book correctly 1.Send request to edit the order of the list 2. Watch for response and check database	System will be able to delete books from a custom list and change the order of the books in the custom list	PASS
TC - 006	US 12	Automatically Add Recommendations to a List	When the user requests recommendations, they will automatically be stored in a list titled "Recommendations" with no additional input from the user	-User with preferences	1. Send request to ML API to generate recommendations 2. Watch for response that recommendations were sent to the backend 3. Check database that the new list was created and all books were added	When recommendations are sent to the backend, then a list will automatically be created	PASS
TC - 007	US 19	Account Registration	When the user registers for an account, the registration will happen correctly and the user will be stored	-Test email and password	1. Use front end to initiate account registration 2. Check for confirmation from front end to user 3. Check in firebase that user was created successfully	User will be successfully created and stored in Firebase	PASS

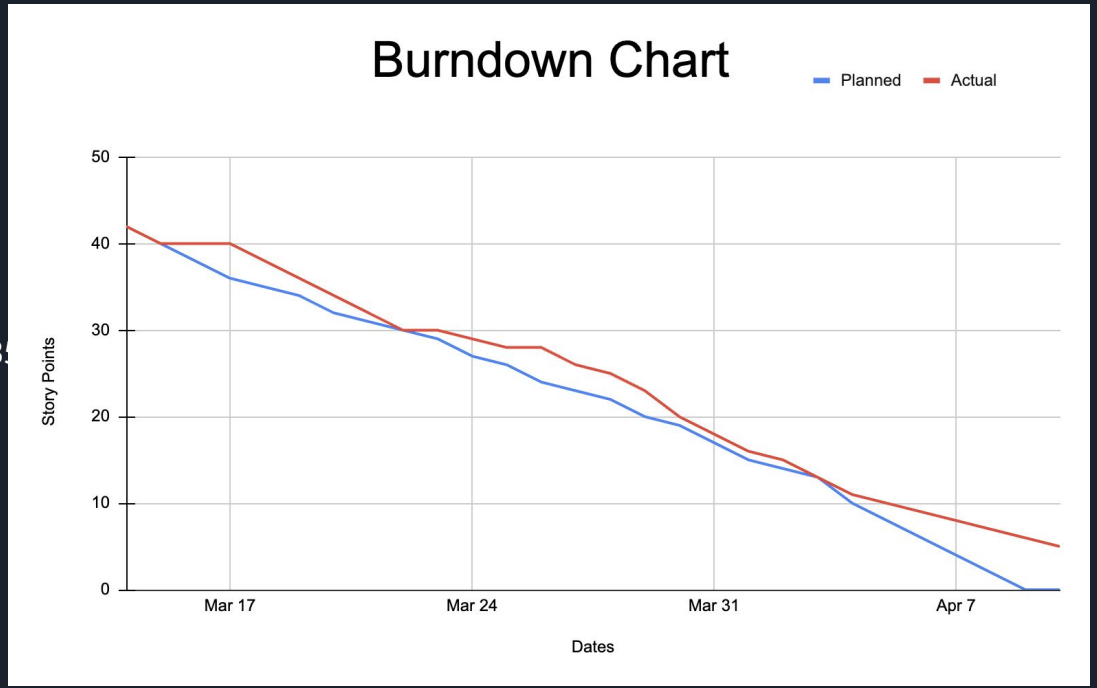
TC - 008	US 21	Password Recovery	If the user forgets their password, there should be an easy to use password recovery feature	-Created user to test recovery	1. Initiate password recovery with a created user 2. Follow recovery process 3. Login with new password	User will be able to successfully login with the new password	PASS
TC - 009	US 22	Log out Functionality	The user should be able to logout from the site	-Created user to test log out	1. Logout of site when signed it 2. Make sure there is no access to the user's account unless they login again	User will be able to end their session and log out completely	PASS
TC - 010	US 13	View book details	The user should be able to view the details of a book	-Created User -Created Book	1. When viewing recommendations or a list of books click on a book 2. Make sure a page populates with all details of book 3. See if description is available and all information matches database	User will be able to see all details for any book stored in the database	PASS
TC - 011	US 14	Add books to Shopping Cart	The user should be able to add items to their shopping cart	-Created User -Created Book	1. Select book 2. Use add to cart button to add the item to a cart 3. Check database to see if everything was stored correctly 4. Log out of user and log back in to check persistence of cart	User will be able to add items to a shopping cart and it will persist in the database	PASS

Sprint 3 Metrics and Charts

Team Velocity: 30 completed story points

Average Velocity: 29 story points

Completed Committed Ratio for Sprint 3: 30/3
- 85%



Sprint 3 Retrospective

Sprint 3 Retro

What went well

- Team coordination (+5)
- all the tasks are being completed on time (+5)
- Teammates helping each other in resolving errors (+3)
- Features of front end are covered (+3)
- development of ml api is successful (+3)
- completing the tasks assigned (+3)
- ML tasks are finished on time (+3)
- Backend team coordinated well (+2)
- Good collaboration of machine learning team (+2)

What can be improved

- Keeping each other updated on what we are working on (+4)
- git commits (+3)
- Get everyone involved in the meetings (+3)
- Time management (+2)
- Git activity (+2)
- Git commits (+2)
- Double checking the presentation for issues (+2)
- more collab coding (+1)

Action Items

- The team will use github more actively and all team members try to commit (+0)
- Every week update other team members on what work is being accomplished (+0)
- All team members should double check the presentation and wikipage against the checklists (+0)

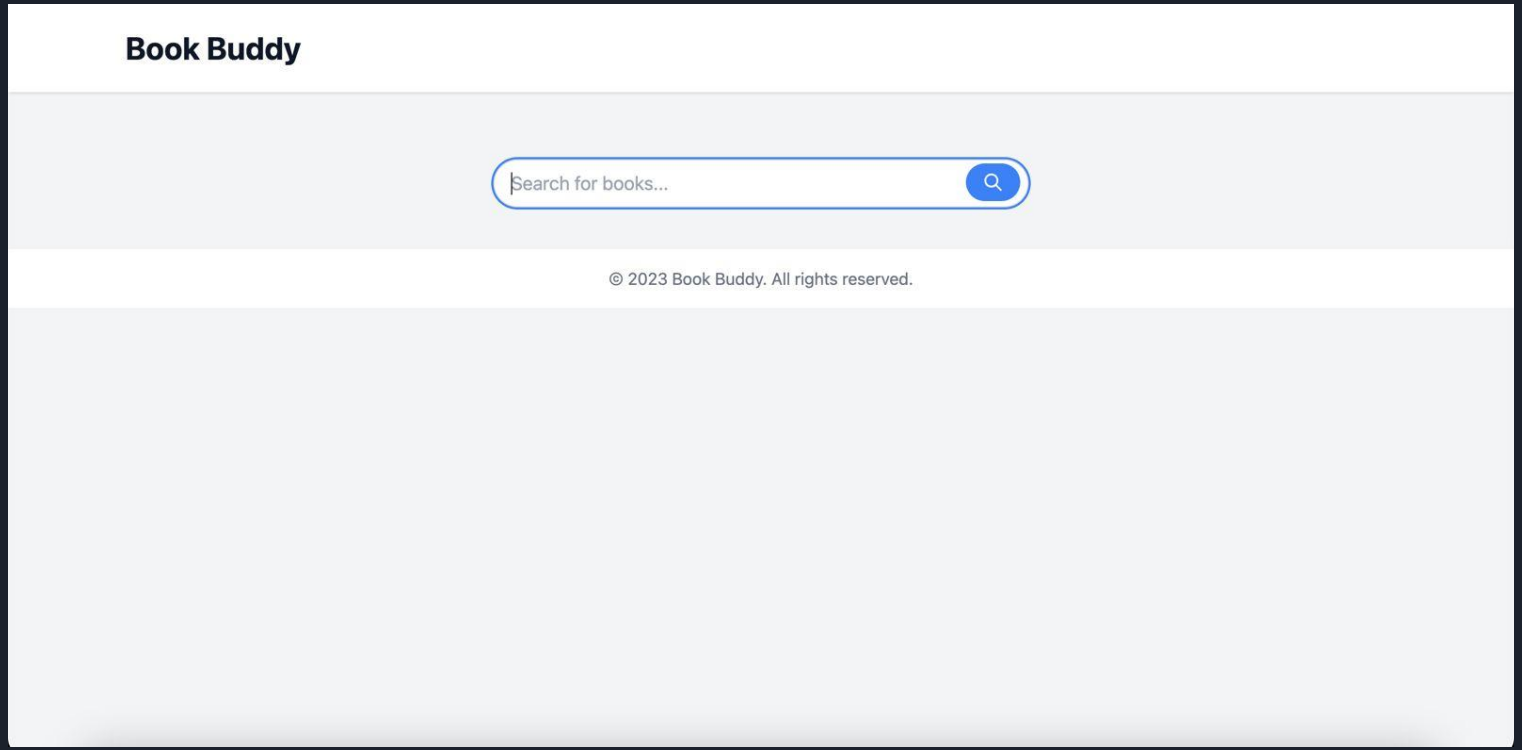
Sprint 4 Plans

Sprint 4 Stories

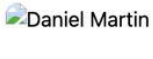
User Story ID	Sprint 4 Stories	POINTS	STATUS
16	As a busy reader, I want to have a smooth and secure checkout process, including entering my payment and shipping information, so that I can complete my purchase.	5	Not Started
17	As a busy reader, I want to view my order history, including past purchases and order status, so that I can track my purchases.	3	Not Started
4	As a passionate reader, I want the system to generate recommendations that promote diversity and highlight authors from different backgrounds	8	Not Started
5	As a passionate reader, I want to be able to adjust my likes and dislikes so that I can refine the recommendations.	3	Not Started
7	As a new reader with no reading history, I want to specify my preferred genre so that recommendations are more tailored to my interests.	5	Not Started
18	As a busy reader, I want the option to use different payment methods when purchasing books, so that I am not limited to only one.	3	Not Started
20	As a passionate reader, I want to be able to register for an account with third party providers (Facebook, Google, etc), so that I can access the application and not have to remember another password.	3	Not Started
15	As a busy reader, I want to view and manage the items in my shopping cart, including adding, removing, and updating quantities, so that I can finalize my purchase.	5	Not Started
8	As a new reader with no reading history, I want to explore recommended books with brief descriptions or summaries to help me decide, so I can read more.	3	Not Started
1	As a passionate reader, I want to input my likes and dislikes into the system, so that the recommendation system can take them into account	2	In Progress
6	As a new reader with no reading history, I want to receive basic recommendations based on popular and new authors in genres I like , so that I can read more.	3	In Progress
		43	

Project Demo for Sprint 3

Demo Screenshots



Recommended Books



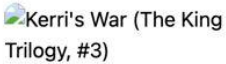
Daniel Martin

Author: John Fowles

BOOKS

Genres: ['Fiction', 'Classics', 'Literature', 'Literary Fiction', 'Novels', 'British Literature', 'Contemporary']

Like



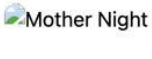
Kerri's War (The King Trilogy, #3)

Author: Stephen Douglass

Librarian Note: Alternate Cover Edition for ASIN:B00DEKRUQ0.

Genres: ['Thriller', 'Romance', 'Crime', 'Amazon']

Like



Mother Night

Author: Kurt Vonnegut Jr.

Librarian note: Alternate cover edition for this ISBN can be found here.Mother Night is a...

Genres: ['Fiction', 'Classics', 'Historical Fiction', 'War', 'Literature', 'Novels', 'Humor']

Like

The Library Book

Author: Susan Orlean

On the morning of April 29, 1986, a fire alarm sounded in the Los Angeles Public...

Genres: ['Nonfiction', 'History', 'Books About Books', 'Audiobook', 'True Crime', 'Adult', 'Historical']

Like

Book Buddy

ha

Harry Potter and the Philosopher's Stone (Harry Potter, #1)
Author: J.K. Rowling
Add

The Diary of a Young Girl
Author: Anne Frank
Add

The Little Prince
Author: Antoine de Saint-Exupéry
Add

The Great Gatsby
Author: F. Scott Fitzgerald
Add

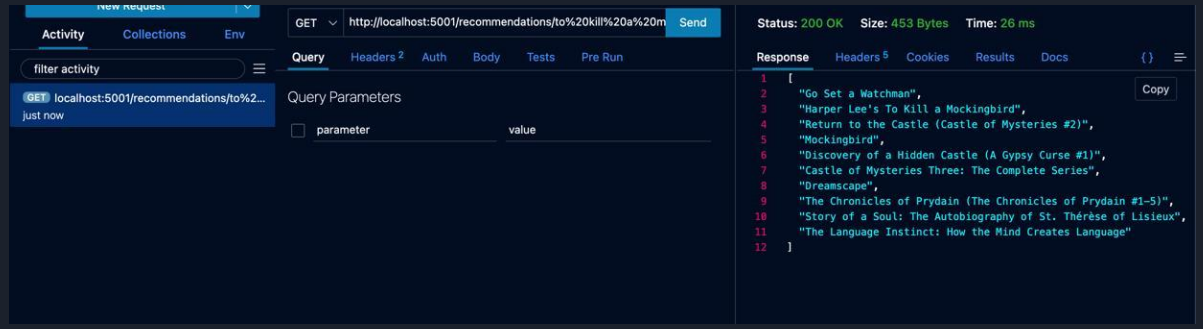
The Catcher in the Rye
Author: J.D. Salinger
Add

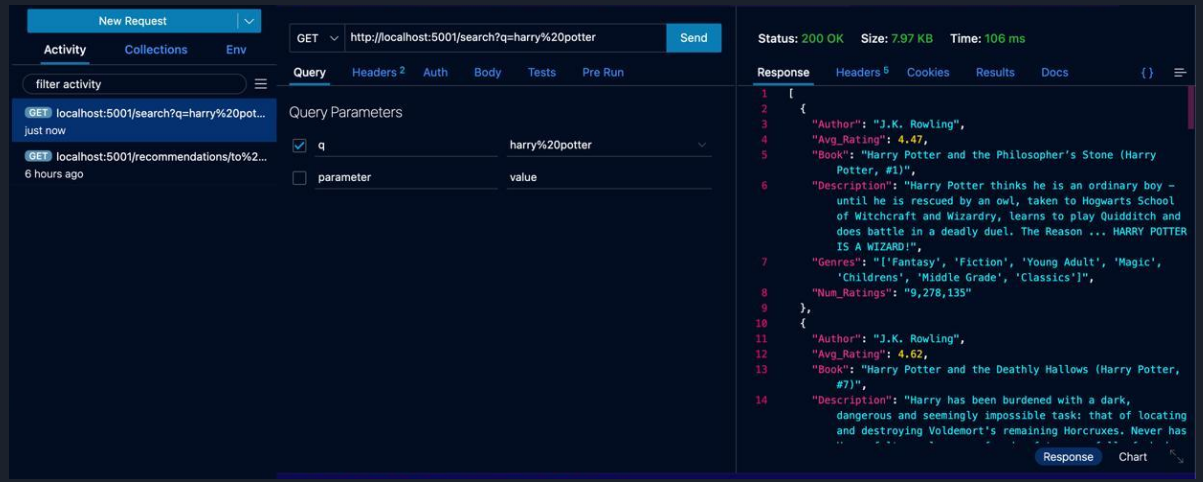
Machine Learning API

We have two routes in our ML API

We can generate recommendations based off a book

We can also search for a book in our dataset





Backend API - User Management

Code Snippet of the User Controller

Shows all CRUD routes for User Management

Using User Data Transfer Object to display user details

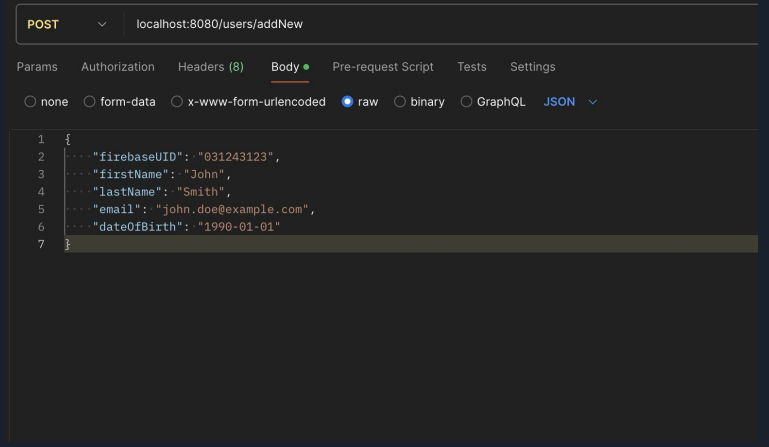
```java
@RestController
@RequestMapping("/users")
@CrossOrigin("*")

public class UserController {
    private final UserRepository uRepository;
    private final UserService uService;

    public UserController(UserRepository uRepository, UserService uService){
        this.uRepository = uRepository;
        this.uService = uService;
    }

    @GetMapping("/all")
    public List<User> all(){
        return uRepository.findAll();
    }

    @PostMapping("/addNew")
    public User newUser(@RequestBody User newUser) {
        System.out.println(newUser);
        return uService.addNewUser(newUser);
    }

    @GetMapping("/{userId}")
    public UserDTO findUser(@PathVariable Long userId) {
        return uService.getUserDetails(userId);
    }

    @PutMapping("/{userId}")
    public ResponseEntity<User> updateUser(@PathVariable Long userId, @RequestBody User updatedUserDetails) {
        User updatedUser = uService.updateUser(userId, updatedUserDetails);
        return ResponseEntity.ok(updatedUser);
    }

    @DeleteMapping("/{id}")
    public ResponseEntity<Long> deleteUser(@PathVariable Long id){
        Optional<User> userOptional = uRepository.findById(id);
        if (userOptional.isPresent()) {
            uRepository.deleteById(id);
            return ResponseEntity.ok(id);
        } else {
            return ResponseEntity.notFound().build();
        }
    }
```

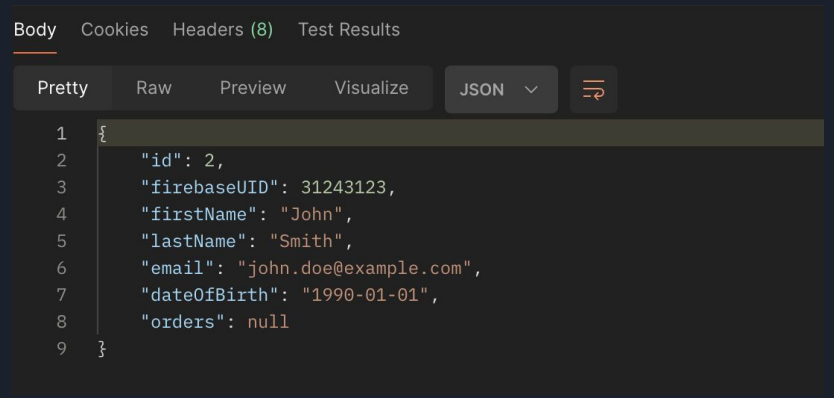
User Creation Route Testing

Route testing in Postman

The route take a new user in the form of a user object in the request body

When the user is stored in the database a unique id is generated and the new user object is returned to the client

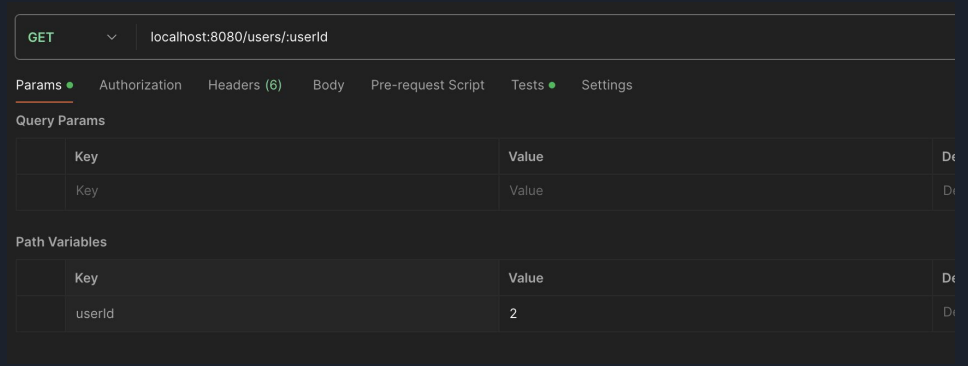


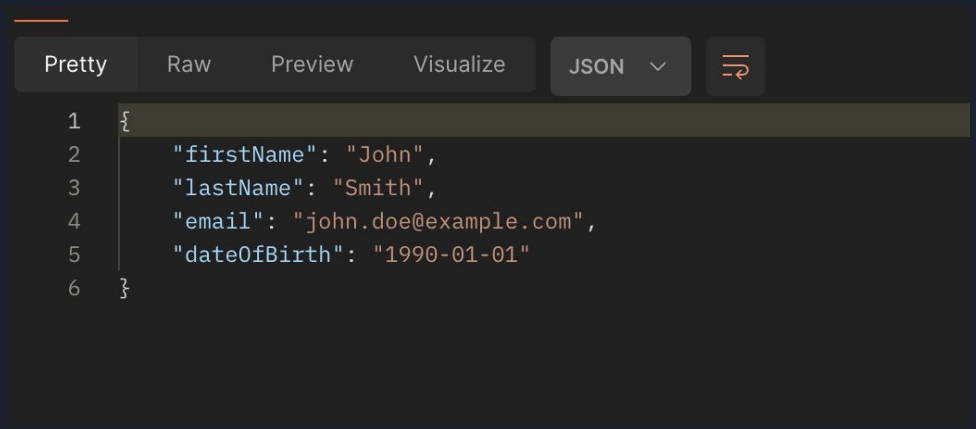


GET User Route Testing

The route takes one path variable which is the user id

This route will be used for populating basic details about the user. When using this route we return a User Data Transfer Object which removes the FirebaseUID and the Database ID since this information will never be displayed to the user





Backend API - Book Collection Management

These are the CRUD routes for Book Collections

These allow the user to create, edit and delete custom book collections

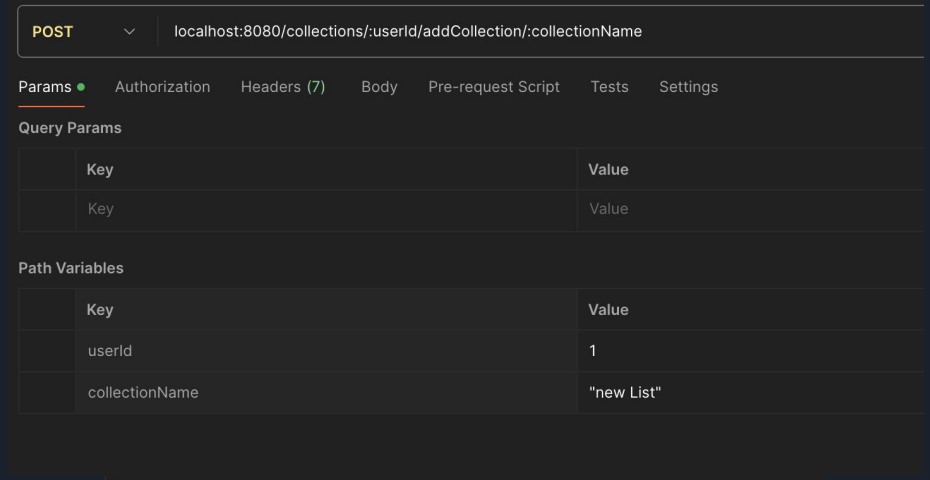
```java
@RestController
@RequestMapping("/collections")
public class BookCollectionController {

    private final BookCollectionRepository bCRepository;
    private final UserRepository uRepository;
    private final BookCollectionService bCService;

    BookCollectionController(BookCollectionRepository bCRepository, UserRepository uRepository, BookCollectionService bCService){
        this.bCRepository = bCRepository;
        this.uRepository = uRepository;
        this.bCService = bCService;
    }

    @GetMapping("/{userId}/{collectionId}")
    public BookCollection getCollection(@PathVariable Long userId, @PathVariable Long collectionId) {
        Optional<User> userOptional = uRepository.findById(userId);
        if(userOptional.isPresent()){
            User user = userOptional.get();
            return user.findCollectionById(collectionId);
        }
        else throw new UserNotFoundException(userId);
    }

    @DeleteMapping("/{userId}/{collectionId}")
    public boolean deleteCollection(
            @PathVariable Long userId,
            @PathVariable Long collectionId) {
        boolean deleted = bCService.deleteCollection(userId, collectionId);
        return deleted;
    }

    @PostMapping("{userId}/addCollection/{name}")
    public BookCollection addCollection(@PathVariable Long userId, @PathVariable String name){
        return bCService.createCollection(userId, name);
    }

    @PutMapping("{userId}/{collectionId}/rename/{newName}")
    public BookCollection updateCollection(@PathVariable Long userId, @PathVariable Long collectionId, @PathVariable String newName){
            return bCService.renameCollection(userId, collectionId, newName);
    }

    @PutMapping("{userId}/{collectionId}/addBook/{bookId}")
    public BookCollection addBook(@PathVariable Long userId, @PathVariable Long collectionId, @RequestBody Book book){
        return bCService.addBook(userId, collectionId, book);
    }
```

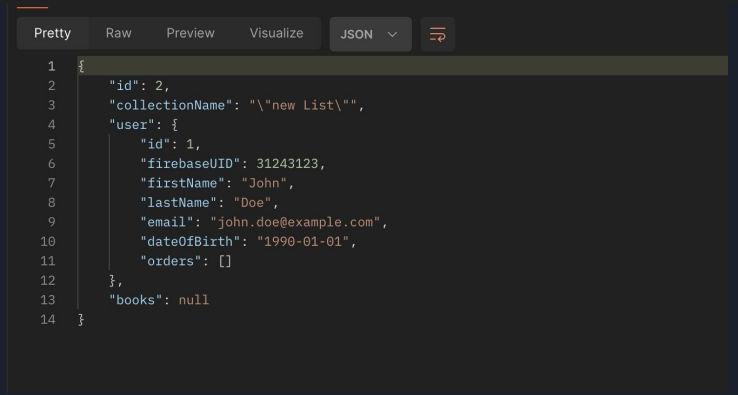
Create Collection Route Testing

This route allows the user to create a new book collection.

There are two path variables: user id and the name the user wants to give to the collection

The system sends the newly created collection along with the user who created it back to the client for display





WikiPage Link

https://github.com/htmw/2024S-Code-Avengers/wiki

Application Demo