

# Sprint 1 Report

CHAPTR - Book tracking platform

Submitted by:  
Fatima Hameed 22k-4656  
Hania Fayyaz 22L-6941  
Ziyad Hisham 22L-6855

## **Product Backlog (All 25)**

1. User Account Creation
2. User Authentication
3. Reading List Addition
4. Reading List Removal
5. Reading Progress Tracking
6. Book Club Creation
7. Book Club Joining
8. Book Club Posting
9. Search for Books by Genre
10. Reading Position Bookmarking
11. Book Reviewing
12. Book Rating
13. Author Following
14. Book Announcement (Author)
15. Update User Accounts
16. Manage My Books
17. Follow Authors
18. See a Calendar of Upcoming Book Releases
19. Follow Other Readers
20. Create Reading Challenges
21. Participate in reading Challenges
22. View Monthly Reports
23. See an Annual Summary of Reading Habits
24. Block Users
25. Add release dates (admin)

## **Sprint backlog**

- User Account Creation
- User Authentication
- Reading List Addition
- Reading List Removal
- Reading Progress Tracking
- Book Club Creation
- Book Club Joining
- Book Club Posting
- Search for Books by Genre

## **Modules**

- User Management Module
- Reading List Module
- Book Club Module
- Book Discovery Module

# User stories for sprint 2

## 1. Reading Position Bookmarking

*User Story:* As a reader, I want to bookmark my current reading position in a book, so that I can easily resume reading later.

*Sub-User Stories:*

- As a reader, I want to view all my bookmarks for a book, so that I can jump to different sections.
- As a reader, I want to delete old bookmarks, so that my list stays organized.

## 2. Book Reviewing

*User Story:* As a reader, I want to write a detailed review for a book I've read, so that I can share my thoughts with the community.

*Sub-User Stories:*

- As a reader, I want to edit or delete my reviews, so that I can correct mistakes or update my opinions.

## 3. Book Rating

*User Story:* As a reader, I want to rate books on a 5-star scale, so that I can quickly express my satisfaction level.

*Sub-User Stories:*

- As a reader, I want to see the average rating of a book, so that I can gauge its popularity.

## 4. Author Following

*User Story:* As a reader, I want to follow my favorite authors, so that I can get updates on their new releases.

*Sub-User Stories:*

- As a reader, I want to see a list of authors I follow, so that I can check their latest activity.

## 5. Book Announcement (Author)

*User Story:* As an author, I want to announce upcoming books, so that my followers are notified.

*Sub-User Stories:*

- As an author, I want to edit or delete my announcements, so that I can update release details.

## 6. Update User Accounts

*User Story:* As a user, I want to update my account details (e.g., username, email, profile picture), so that my profile stays current.

Sub-User Stories:

- As a user, I want to change my password securely, so that I can keep my account safe.

## 7. Manage My Books

*User Story:* As a reader, I want to manage my reading list (e.g., mark books as "Read," "Currently Reading," or "Want to Read"), so that my progress is tracked accurately.

Sub-User Stories:

- As a reader, I want to filter books by status, so that I can organize my reading goals.

## 8. Set Daily Reading Goals

*User Story:* As a reader, I want to set Daily reading goals to motivate me to read more everyday

## 9. Calendar of Upcoming Book Releases

*User Story:* As a reader, I want to view a calendar of upcoming book releases, so that I can plan my future reading.

Sub-User Stories:

- As a reader, I want to filter releases by genre, so that I can find books I'm interested in.

## Technical details for sprint 1 user stories:

### 1. User Account Creation

*Backend:*

- Create User model with fields: userId, username, email, password (hashed), role (reader/author/admin).
- Implement POST /api/users/register endpoint with validation (unique email, strong password).
- Use JWT for token generation upon successful registration.

*Frontend:*

- Build a registration form (React) with fields: username, email, password, confirm password.
- Add client-side validation (e.g., email format, password strength).
- Redirect to login page after successful registration.

*Database:*

- MongoDB collection: users

Sub-User Story: Prevent Duplicate Accounts

- Backend: Add middleware to check for existing email before registration.
- Frontend: Display error message if email is already registered.

## 2. User Authentication

*Backend:*

- Implement POST /api/users/login endpoint (validates credentials, returns JWT).
- Add session timeout (e.g., JWT expires in 24 hours).
- Secure routes using middleware (authMiddleware.js).

*Frontend:*

- Login form with email/password fields.
- Store JWT in localStorage.
- Redirect to dashboard on successful login.

*Database:*

- Verify password against hashed value in users collection.

Sub-User Story: Handle Invalid Credentials

- Backend: Return 401 Unauthorized for invalid credentials.
- Frontend: Show error message (e.g., "Invalid email/password").

## 3. Reading List Addition

*Backend:*

- Create ReadingList model: bookId, userId, status ("Want to Read"/"Currently Reading").
- Add POST /api/reading-list endpoint to add books.

*Frontend:*

- "Add to Reading List" button on book search results.
- Dropdown to select reading status.

*Database:*

- MongoDB collection: reading\_lists (compound index on userId and bookId).
- Sub-User Story: Sync Across Devices
- Use WebSocket or polling to update reading list in real-time.

## 4. Reading List Removal

Backend:

- Add DELETE /api/reading-list/:bookId endpoint.

Frontend:

- "Remove" button in reading list view with confirmation modal.

Database:

- Soft delete (set isActive: false) or hard delete from reading\_lists.

## 5. Reading Progress Tracking

Backend:

- Extend ReadingList model with progress (e.g., currentPage, percentage).
- Add PATCH /api/reading-list/:bookId/progress to update progress.

Frontend:

- Progress input form (e.g., slider for percentage, numeric input for pages).
- Visual progress bar on book cards.

Database:

- Update reading\_lists documents with progress data.

## 6. Book Club Creation

Backend:

- Create BookClub model: clubId, name, description, createdBy (userId).
- Implement POST /api/book-clubs endpoint.

Frontend:

- Form for club name, description, and privacy settings (public/private).

Database:

- Collection: book\_clubs with reference to users for ownership.

## 7. Book Club Joining

Backend:

- Add POST /api/book-clubs/:clubId/join endpoint.
- Store members in book\_clubs.members[] array (user IDs).

Frontend:

- "Join Club" button on club details page.

Database:

- Update book\_clubs document with new member.

## 8. Book Club Posting

Backend:

- Create Post model: postId, content, authorId, clubId, timestamp.
- Add POST /api/book-clubs/:clubId/posts endpoint.

Frontend:

- Post creation form (textarea + submit button).
- Display posts in a feed (reverse chronological order).

Database:

- Collection: posts with references to users and book\_clubs.

## 9. Search for Books by Genre

Backend:

- Integrate third-party API (e.g., Google Books) or seed local books collection.
- Implement GET /api/books?genre={genre} endpoint.

Frontend:

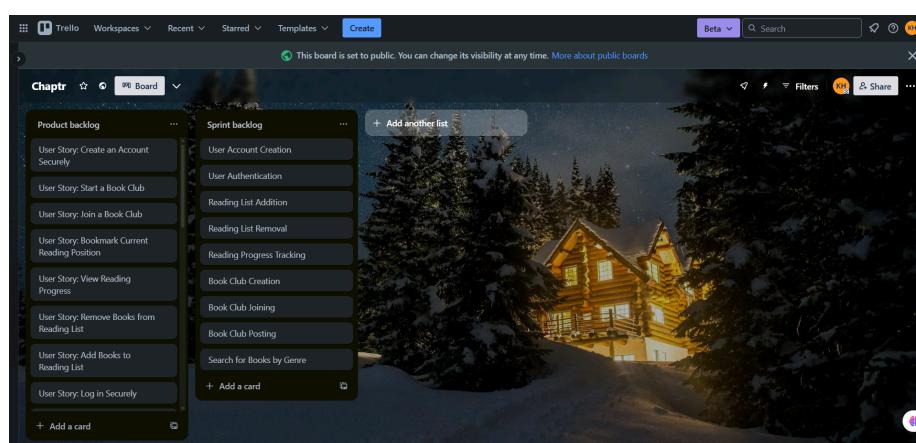
- Search bar with genre dropdown filter.
- Display results in a grid (book cover, title, author).

Database:

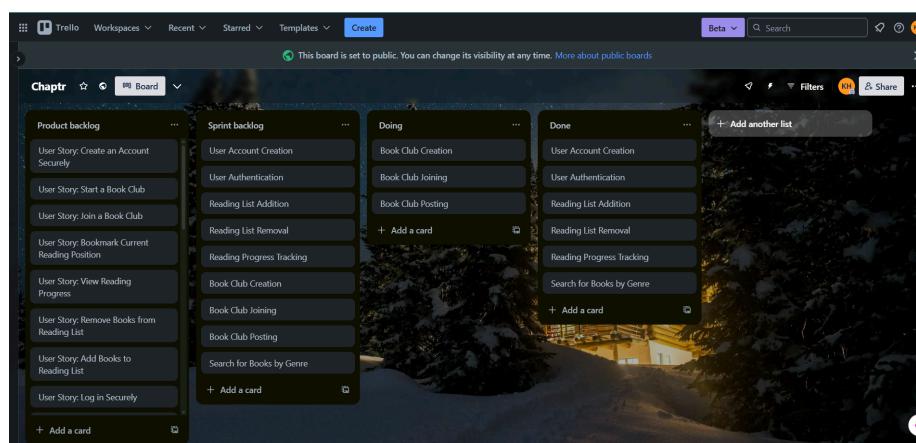
- Collection: books with genre field (indexed for faster queries).

## Trello board

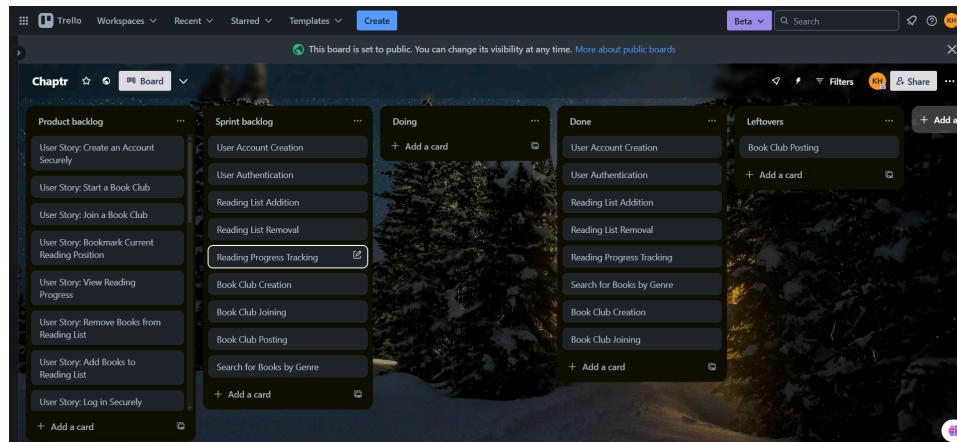
### Beginning



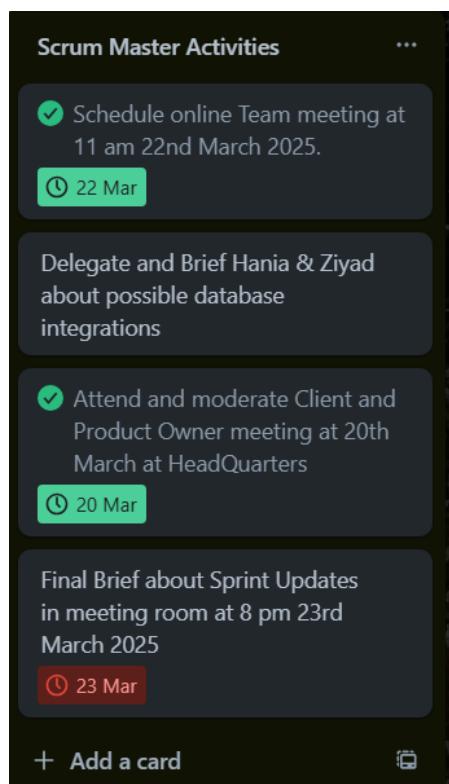
### Middle



End

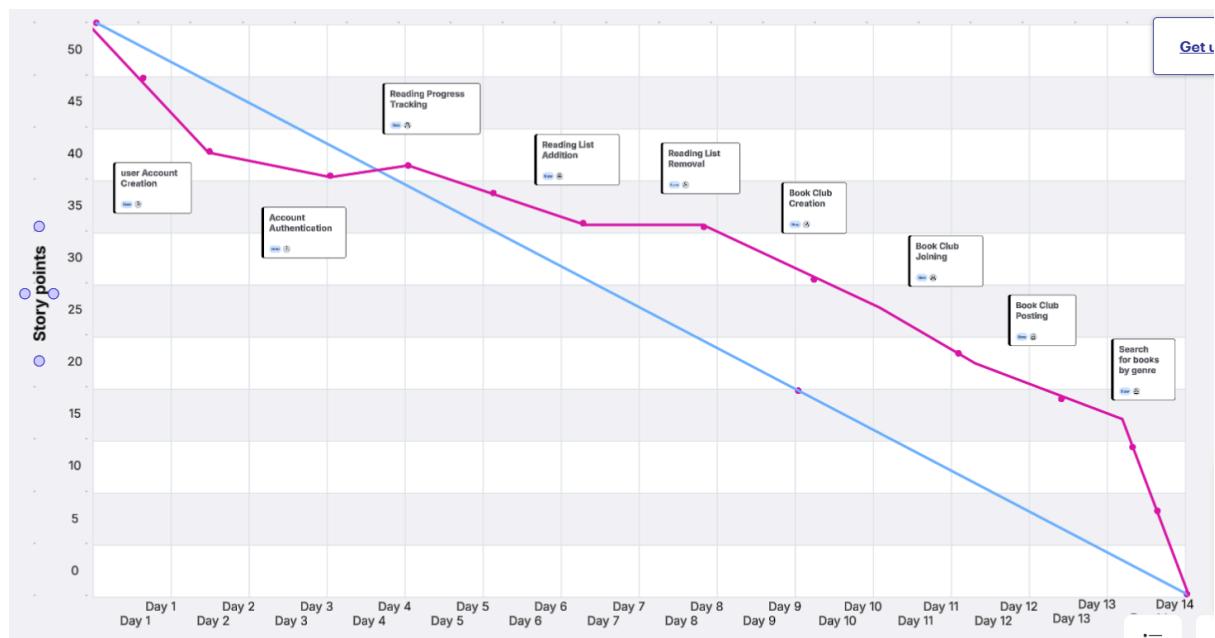


## Scrum Master Activities



## BURNDOWN CHARTS:

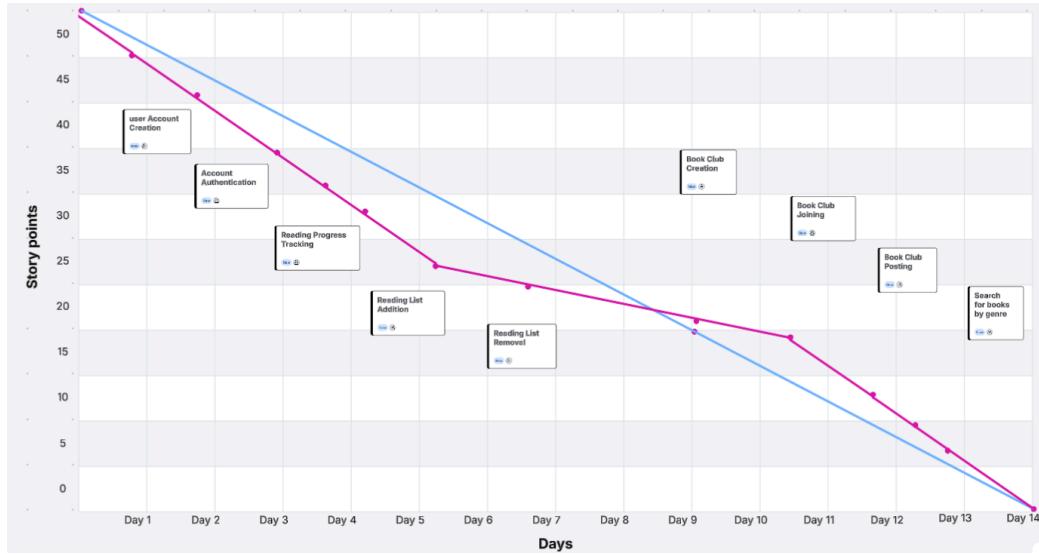
Day 3



DAY 4:



## Day 7



## Day 9



## Day 11



# Weak Equivalent Class Partitioning

## 1. Create an Account Securely

User Story	Input	Valid Partitions	Invalid Partitions
Create an Account Securely	Email	Valid email format (e.g., user@domain.com)	Invalid email format (e.g., user@, user.domain), Empty email
Create an Account Securely	Password	Valid password (alphanumeric)	Empty password
Create an Account Securely	Username	Valid username (alphanumeric)	Empty username

## 2. Log in Securely

User Story	Input	Valid Partitions	Invalid Partitions
Log in Securely	Email	Registered email	Unregistered email, Invalid email format, Empty email
Log in Securely	Password	Correct password for email	Incorrect password, Empty password

## 3. Add Books to Reading List

User Story	Input	Valid Partitions	Invalid Partitions
Add Books to Reading List	User Login Status	Logged in	Not logged in
Add Books to Reading List	Book Selection	Valid book selected	No book selected

## 4. Remove Books from Reading List

User Story	Input	Valid Partitions	Invalid Partitions
Remove Books from Reading List	User Login Status	Logged in	Not logged in

Remove Books from Reading List	Book in List	Book exists in list	Book not in list
--------------------------------	--------------	---------------------	------------------

## 5. View Reading Progress

User Story	Input	Valid Partitions	Invalid Partitions
View Reading Progress	Reading Data	At least 1 page read	No pages read, negative pages read

## 6. Join a Book Club

User Story	Input	Valid Partitions	Invalid Partitions
Join a Book Club	Book Club Existence	Club exists	Club does not exist

## 7. Start a Book Club

User Story	Input	Valid Partitions	Invalid Partitions
Start a Book Club	Club Name	Valid name	Empty name
Start a Book Club	Club Description	Valid description	Empty description
Start a Book Club	Club Cover Image	Valid image (jpg, png)	Empty image
Start a Book Club	Club Tags	Valid tags	Empty tags

## 8. Post in Book Clubs

User Story	Input	Valid Partitions	Invalid Partitions
Post in Book Clubs	Membership Status	Member of club	Not a member
Post in Book Clubs	Post Content	Valid text (any non-empty text)	Empty text

## 9. Search for Books by Genre

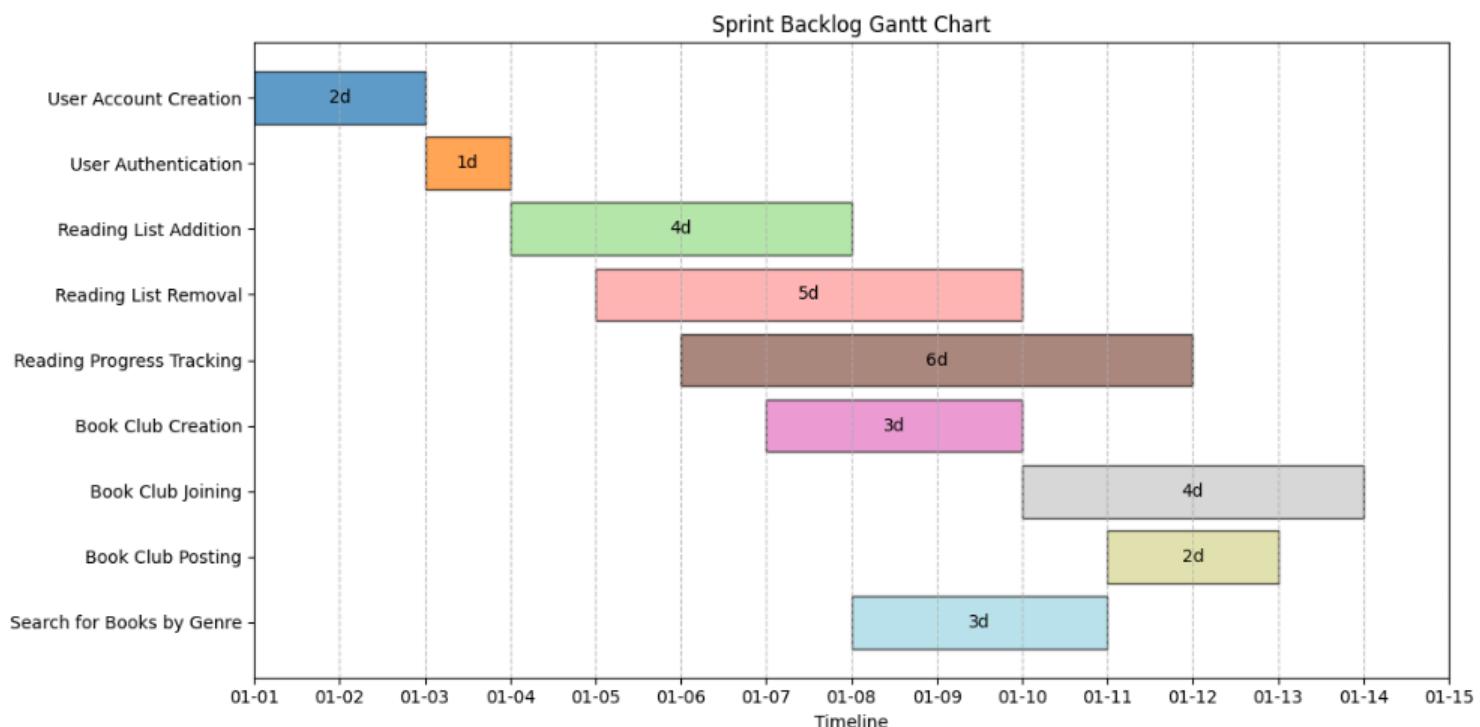
User Story	Input	Valid Partitions	Invalid Partitions
Search for Books by Genre	Genre Input	Valid genre (e.g., Fiction, Non-Fiction)	Empty genre

---

## Number of Test Cases

User Story	Inputs	WECT Test Cases	SECT Test Cases
Create an Account Securely	Email (3), Password (2), Username (2)	3	12
Log in Securely	Email (4), Password (3)	4	12
Add Books to Reading List	Login Status (2), Book Selection (2)	2	4
Remove Books from Reading List	Login Status (2), Book in List (2)	2	4
View Reading Progress	Reading Data (3)	3	3
Join a Book Club	Club Existence (2), User Action (2, assumed)	2	4
Start a Book Club	Club Name (2), Club Description (2), Club Cover Image (2), Club Tags (2)	2	16
Post in Book Clubs	Membership Status (2), Post Content (2)	2	4
Search for Books by Genre	Genre Input (2)	2	2

## Gantt chart



# Sprint Project Planner

Sprint Duration: 14 Days

---

## 1. Sprint Backlog & Task Breakdown

Task	Start Day	Duration (Days)	Dependencies	Completion Criteria
User Account Creation	Day 1	2	None	Users can create accounts with email/password
User Authentication	Day 3	1	User Account Creation	Secure login & session management
Reading List Addition	Day 4	4	None	Users can add books to reading lists
Reading List Removal	Day 5	5	Reading List Addition	Users can remove books from lists
Reading Progress Tracking	Day 6	6	None	Users can track reading % per book
Book Club Creation	Day 7	3	None	Users can create book clubs
Book Club Joining	Day 10	4	Book Club Creation	Users can join existing clubs
Book Club Posting	Day 11	2	Book Club Joining	Users can post in club discussions
Search for Books by Genre	Day 8	7	None	Users can filter books by genre

---

## 2. Key Milestones

- Day 3: Authentication system complete
- Day 7: Reading List features (Add/Remove) functional
- Day 10: Book Club

- Day 14: Sprint review & demo
- 

### 3. Dependencies & Risks

Dependencies:

- Authentication requires Account Creation first.
- 

### 4. Daily Standup Focus

- Days 1–3: Account system progress
  - Days 4–7: Reading List & Progress Tracking
  - Days 8–14: Book Clubs & Search
- 

### 5. Review & Demo Plan

- Demo Features:
    - User registration + authentication flow
    - Reading List CRUD operations
    - Book Club creation/joining + posting
    - Genre-based search
- 

Notes:

- Adjust durations/owners based on team capacity.

**Github url: [https://github.com/haniafayyaz/Chaptr\\_](https://github.com/haniafayyaz/Chaptr_)**

 Chaptr\_ Public

[main](#) [1 Branch](#) [0 Tags](#) [Go to file](#) [t](#) [Add file](#) [Code](#)

 **fatimahameed-yay** updated UI 6d3b5be · 12 hours ago  26 Commits

 .vscode	updated UI	12 hours ago
 backend	updated UI	12 hours ago
 frontend	updated UI	12 hours ago
 node_modules	updated UI	12 hours ago
 README.md	updated UI	12 hours ago
 package-lock.json	updated UI	12 hours ago
 package.json	updated UI	12 hours ago