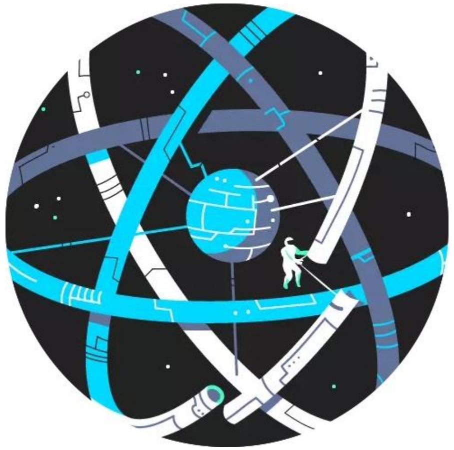
xGame P1

Team ReaKt (t02)



Kassidy Barram

Matt Young

Darin Harter

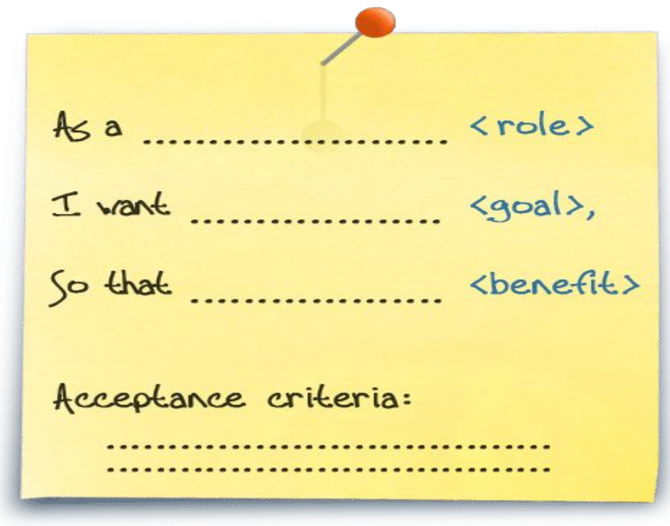
Lani Brooks

April Kelly

Aaron Lawrence

User Stories

- Formulated our user stories by using the INVEST criteria.
- Using acceptance criteria, we then made our tasks.
- Our User Stories


As a .... <role>
I want .... <goal>,
So that .... <benefit>
Acceptance criteria:
....
....

User Stories

Playing the game

Description
As a user, I can play the game of chess according to the rules.

Conversation
The game exists on the website, as some kind of virtual chessboard that users can interact with. The game knows which moves are legal and which are illegal, and automatically enforces the rules of chess so that it is impossible for users to make an illegal move. The application knows whose turn it is and does not allow users to make moves out of turn. The application knows when the game is over, and who won.

Acceptance Criteria
- Test that when a user moves a piece the board updates.
- Test that a user can't move out of turn.
- Test that a user can't make an illegal move.
- Test that the system recognizes when the game is over.
- Test that the system knows who won a completed game.

Match invitations

Description
As a user, I can send an invite for other registered users to join and play a match.

Conversation
A player can find an opponent to invite to a game by means of searching or viewing a table of registered users. The player will then be able to send an invitation to their selected opponent and the invitee will then be prompted to either accept or decline the invitation. The inviter will then be notified of the invitee's answer. When an invitation is accepted, a match will be created between the two players.

Acceptance Criteria
- Test if user can send invitations to other users.
- Test that a user gets a notification when they receive an invitation.
- Test that a user can accept an invitation they receive.
- Test that a user can decline an invitation they receive.
- Test that a user gets notified when another user accepts the invitation.
- Test that a user gets notified when another user declines the invitation.
- Test that accepting makes a match.
- Test that a match is only created with the first person who accepts the invitation.

Returning to a game after logging out

Description
As a user, I can return to a game after I log out.

Conversation
What else? Oh right. The state of the matches should be saved in some way, so the user can play whenever she wants. My guess is that users won't be playing the whole time, so for example, a user would make a move whenever is her turn and log out, and after a while she would come back and check if the other player made a move and it's her turn again.

Acceptance Criteria
- Test that a user can login.
- Test that a user can logout.
- Test that a game can be saved.
- Test that the relationship between a specific saved game and a specific user is preserved after log out.
- Test that a game can be loaded from a save.
- Test that a user can return to a game after logging out.
- Test that a user can see a list of active games to return to.

Playing multiple games at once

Description
As a user, I can play multiple games at once.

Conversation
Several games can be played at a time, and those games are unrelated. The games may begin and end at any time, unaffected by the status or participants of any other game.

Acceptance Criteria
- Test that a user be involved in multiple games at once.
- Test that a user's moves in one game does not affect their activity in another game.
- Test that if a user quits one game, they do not leave any other game.
- Test that joining a new game does not affect existing games.
- Test that a user can view a list of all matches they are apart of.

Viewing user profiles

Description
As a user, I can view my own profile or those of the other registered users.

Conversation
A player can select another user's profile to view their game statistics and profile information. This includes a user picture and a short bio section.

Acceptance Criteria
- Test if the user can view other user's profiles.
- Test if the user can view their own profile.
- Test if the user can see their statistics update on their profile.
- Test if the user can see other user's statistics update on their profile.
- Test to see if the database is updating with all information.

Notification of an opponent's move

Description
As a user, I can be notified when my opponents make a move.

Conversation
Asynchronous matches, I think that describes it.The system needs to know when a game is over and should let know the players who won or lost.

Acceptance Criteria
- Test that a user can receive a notification.
- Test that the system can send a notification.
- Test that a win triggers a notification.
- Test that a loss/concession triggers a notification.
- Test that both players are notified.

Navigating the page

Description
As a user, I can navigate the web page between the different sections based on whether I want to play a game, view a users profile, or register/unregister.

Conversation
The user profiles, registration page, and game board is all on a website. Different tabs would allow one to switch between the different features, as long as the user is currently signed in.

Acceptance Criteria
- Test that there are multiple tabs at the top of the web page
- Test that the tabs take you to the correct pages.
- Test that the tabs either don't work or take you to a register prompt if a user attempts to use them without being logged in.

Register/ unregister an account

Description
As a user, I can register and unregister an account.

Conversation
Completing the registration form on the website creates an account for a user, and that account and its information are saved. Registered users are able to log in to their account. Choosing to delete the account causes the account and all of its information to be removed from the database. Deleted accounts cannot be logged into.

Acceptance Criteria
- Test that completing the registration form adds an entry to the Users table in the database.
- Test that logging in to a registered account succeeds.
- Test that the information belonging to a user persists between logging out and logging back in.
- Test that deleting an account removes that entry from the database.
- Test that trying to log in to a deleted account fails.
- Test that creating a new account with the same name/username/email as a deleted account yields a new, blank account

Quitting a game

Description
As a user, I can quit the game at any time.

Conversation
The platform will have an option to quit the game at any time while the game is being played. Once a user decides to quit their current game, a second window will appear confirming whether or not the user wants to quit the game or not. If they select yes, the game ends for both users. If they select no, the game resumes for both users. If the user selects yes, the game will be stored in each user profile as an abandon game.

Acceptance Criteria
- Test that when a user quits the game, they can not revisit the game.
- Test that when a user quits the game, it does not quit any other game they are currently playing, for both players.
- Test that when a user quits the game, it is stored in the database as an abandon game for both players.
- Test that when a user quits the game, is is added to their profile, for both players.

Recording game history and stats

Description
As a user, I can have my statistics related to games I've played recorded.

Conversation
When you finish a game, information related to that game will be stored and displayed on their profile. This information includes the times and dates the game began and ended, who won and who lost the game, and whether the game was abandoned partway through. All this information will be recorded for multiple games, and will all be displayed on a users profile.

Acceptance Criteria
- Test that the start time and date are recorded when a game is started.
- Test that the end date and time are recorded when a game is finished.
- Test that the way the game ends is recorded (who wins and loses, or whether the game was abandoned)
- Test that the game information that is recorded updates on the users profile correctly.
- Test that the information for multiple games can be recorded at the same time.

Viewing the rules

Description
As a user, I can view the rules at any time.

Conversation
A player can easily navigate to the rules at any time whether before, during, or after a game. The rules are very straight forward, easy to understand, and easy to read. A player will be able to read the rules, top to bottom, or have the option to search for specific words and be taken to that specific part of the rules.

Acceptance Criteria
- Test that a user can navigate to the rules while playing the game.
- Test that the rules are legible.
- Test that the search bar takes you to the specified rule.
- Test if the profile is only visible to registered users.

Searching for users

Description
As a user, I can see a list of all registered users or search for a specific user to view or play against.

Conversation
A user can search and filter through all other registered users to find an opponent. The query results may be narrowed down using a partial or exact username. A form of sorting might be permitted to make users easier to find.

Acceptance Criteria
- Test that entering a partial word filters the list of users.
- Test that searching for an exact name returns one result, since names are unique.
- Test that all registered users are listed if there is no search term.
- Test that the results list can be sorted alphabetically or by recent activity.

Updating user profiles

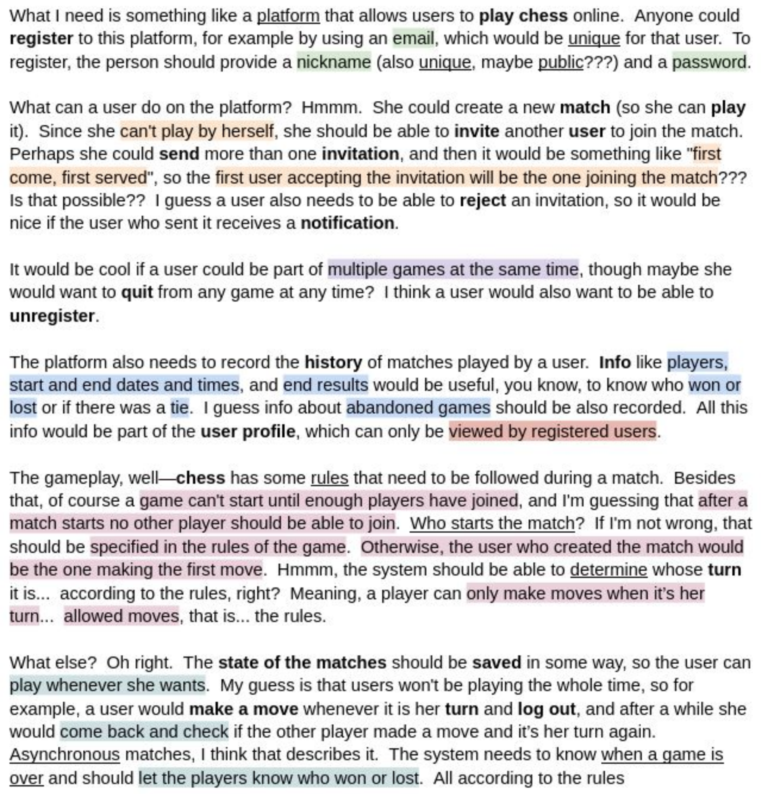
Description
As a user, I can update and personalize my profile.

Conversation
A registered user can visit their own profile and configure it to their liking. Information such as nickname, avatar, and bio can be edited and saved for others to see.

Acceptance Criteria
- Test that a user can only update their own profile information.
- Test that changes persist across page reloads and logouts.
- Test that the user can change the nickname.
- Test that the new nickname must be unique.
- Test that the user can change the pic/ avatar.
- Test that the user can change the bio.
- Test updates with long text inputs.
- Test updates with empty text inputs.
- Test that other users can see the changes (the database is updated).

CRC and Server Design

- We built CRC cards based off of our User Stories.
- Mapped nouns in the conversation to classes in our cards.
- UML Diagram was designed using our CRC Cards.



What I need is something like a platform that allows users to **play chess** online. Anyone could **register** to this platform, for example by using an email, which would be unique for that user. To register, the person should provide a nickname (also unique, maybe public???) and a password.

What can a user do on the platform? Hmmm. She could create a new **match** (so she can **play** it). Since she can't play by herself, she should be able to **invite** another **user** to join the match. Perhaps she could **send** more than one **invitation**, and then it would be something like "first come, first served", so the first user accepting the invitation will be the one joining the match??? Is that possible?? I guess a user also needs to be able to **reject** an invitation, so it would be nice if the user who sent it receives a **notification**.

It would be cool if a user could be part of multiple games at the same time, though maybe she would want to **quit** from any game at any time? I think a user would also want to be able to **unregister**.

The platform also needs to record the **history** of matches played by a user. **Info** like players, start and end dates and times, and end results would be useful, you know, to know who won or lost or if there was a tie. I guess info about abandoned games should be also recorded. All this info would be part of the **user profile**, which can only be viewed by registered users.

The gameplay, well—**chess** has some rules that need to be followed during a match. Besides that, of course a game can't start until enough players have joined, and I'm guessing that after a match starts no other player should be able to join. Who starts the match? If I'm not wrong, that should be specified in the rules of the game. Otherwise, the user who created the match would be the one making the first move. Hmmm, the system should be able to determine whose **turn** it is... according to the rules, right? Meaning, a player can only make moves when it's her turn... allowed moves, that is... the rules.

What else? Oh right. The **state of the matches** should be **saved** in some way, so the user can play whenever she wants. My guess is that users won't be playing the whole time, so for example, a user would **make a move** whenever it is her **turn** and **log out**, and after a while she would come back and check if the other player made a move and it's her turn again. Asynchronous matches, I think that describes it. The system needs to know when a game is over and should let the players know who won or lost. All according to the rules

CRC Cards

Platform	
Navigate to pages	Game
Create games	User
Allows the user to play a game	Profile
Allows user registration/ deregistration	
User is able to view profiles	

User	
Has nickname, password, email	Profile
Allow Login	
Allow Logout	

Profile	
Display user info	GameRecord
Update user info	

Game	
Has moves, turn, gameEnded	ChessBoard
Show game results screen on end	User
Creates a GameRecord when the game ends	GameRecord
	Rules

GameRecord	
Has a winner, loser, start and end time	N/A

Notifications	
Creates a website popup alerting the user on the website.	Game
Sends an email alerting the user	Invite

ChessBoard	
Has ChessPiece[8][8]	ChessPiece
Create the board with pieces at the initial location.	
Updates the board state on successful moves.	

ChessPiece	
Has Color	Pawn
	Rook
	Knight
	Rook
	Queen
	King

Invite	
Has accepted, declined, pending, from user, to user	N/A

Rules	
Check if a game has ended by looking at board state	ChessBoard

Pawn	
Permit or deny a move request	ChessBoard

Knight	
Permit or deny a move request	ChessBoard

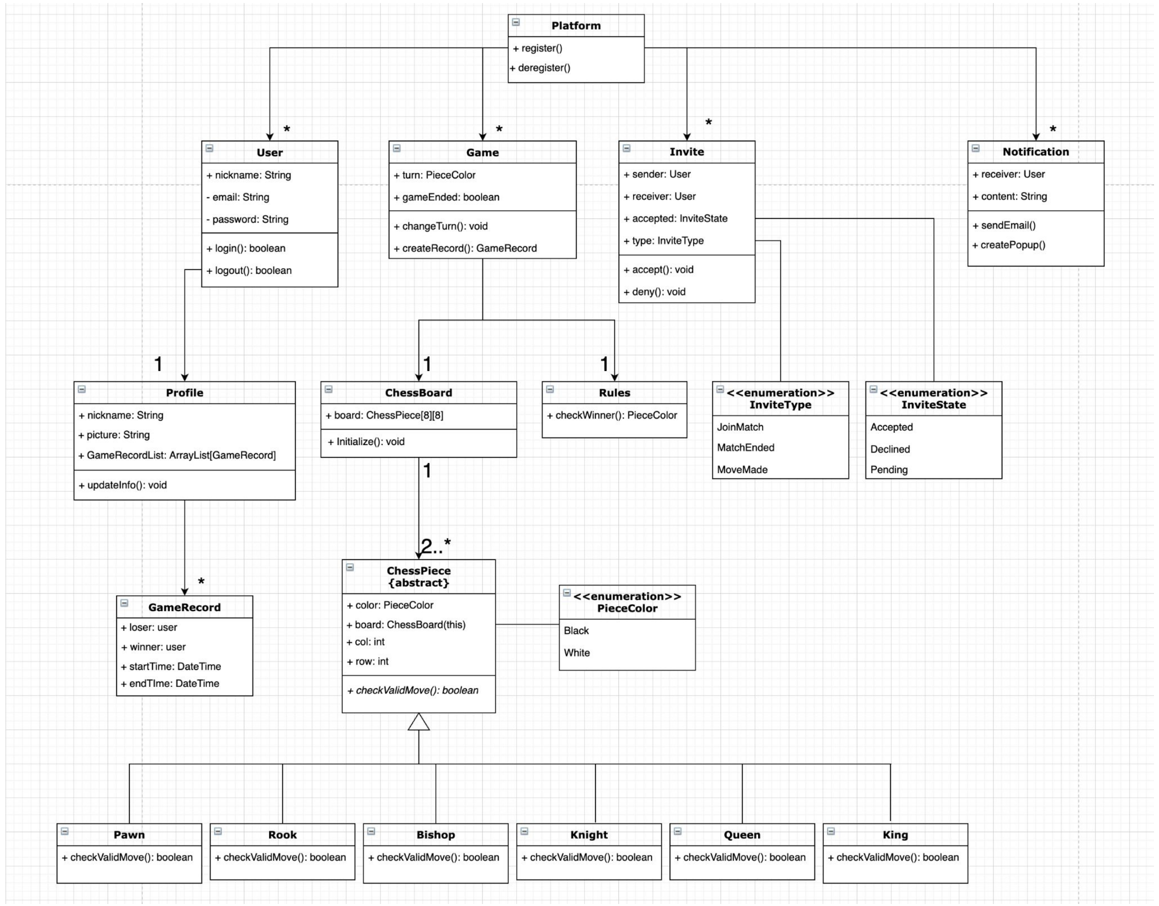
Rook	
Permit or deny a move request	ChessBoard

Bishop	
Permit or deny a move request	ChessBoard

Queen	
Permit or deny a move request	ChessBoard

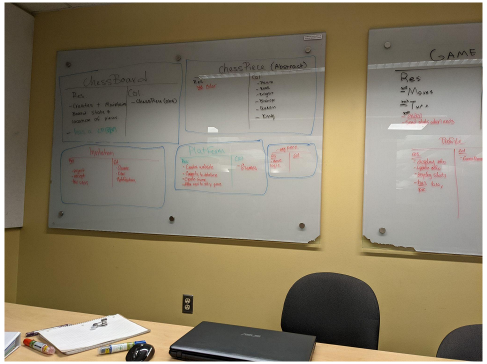
King	
Permit or deny a move request	ChessBoard

Server Class Diagram



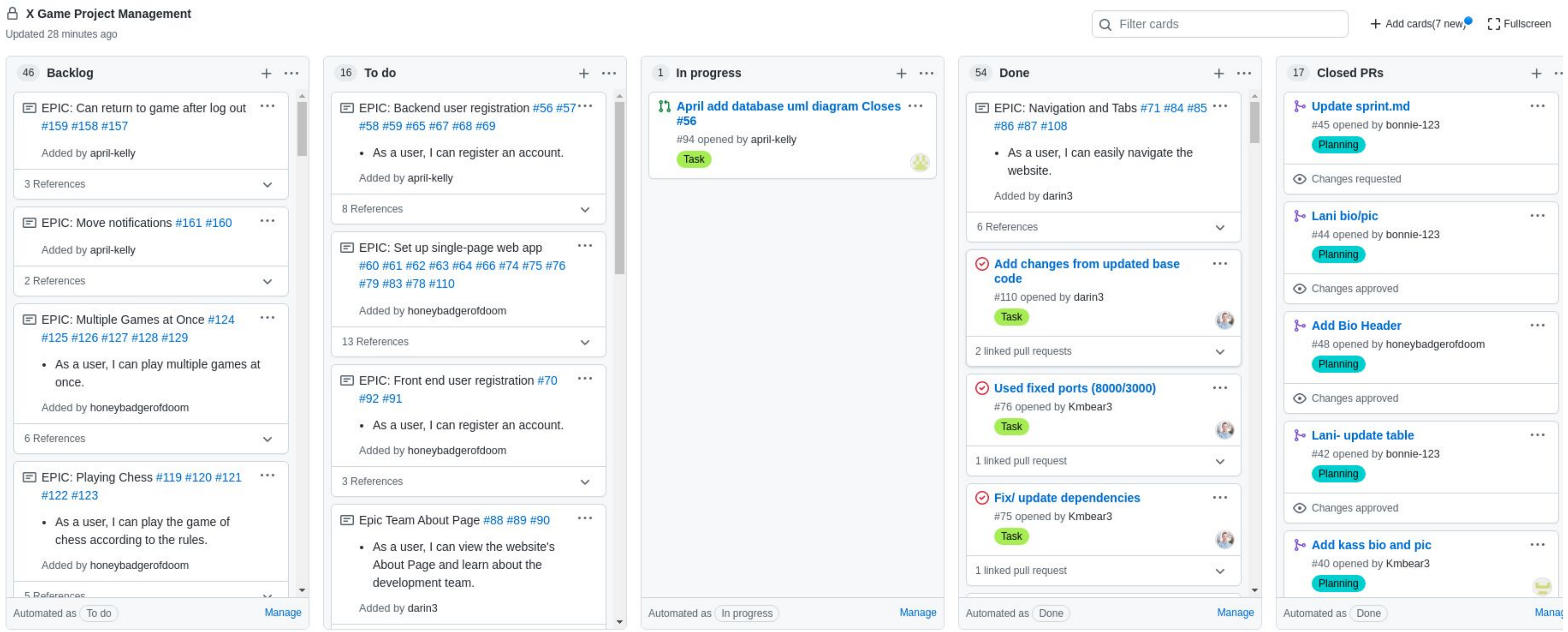
Process and Scrum Ceremonies

- Meetings three days a week
  - M, W, S at 1pm
  - Scrum meetings ~10 minutes
- Preliminary planning meetings at the start
  - Initially focused too much on implementation.
  - Shifted to planning and design based approach.
- Some meetings are designed to be teamwork based
  - Find time for longer meetings.
  - Collaborate on shared documents via Google Drive.
  - Major design planning: hold in-person meetings.



Kanban Planning Progress

- Using GitHub Projects, we created <u>cards</u> to represent <u>Epics</u>, then <u>associated tasks</u> to those cards.



X Game Project Management
Updated 28 minutes ago

Filter cards | + Add cards(7 new) | Fullscreen

46 Backlog

EPIC: Can return to game after log out #159 #158 #157
Added by april-kelly
3 References

EPIC: Move notifications #161 #160
Added by april-kelly
2 References

EPIC: Multiple Games at Once #124 #125 #126 #127 #128 #129
- As a user, I can play multiple games at once.

Added by honeybadgerofdoom
6 References

EPIC: Playing Chess #119 #120 #121 #122 #123
- As a user, I can play the game of chess according to the rules.

Added by honeybadgerofdoom
5 References

Automated as To do — Manage

16 To do

EPIC: Backend user registration #56 #57 #58 #59 #65 #67 #68 #69
- As a user, I can register an account.

Added by april-kelly
8 References

EPIC: Set up single-page web app #60 #61 #62 #63 #64 #66 #74 #75 #76 #79 #83 #78 #110
Added by honeybadgerofdoom
13 References

EPIC: Front end user registration #70 #92 #91
- As a user, I can register an account.

Added by honeybadgerofdoom
3 References

Epic Team About Page #88 #89 #90
- As a user, I can view the website's About Page and learn about the development team.

Added by darin3

1 In progress

April add database uml diagram Closes #56
#94 opened by april-kelly
Task

Automated as In progress — Manage

54 Done

EPIC: Navigation and Tabs #71 #84 #85 #86 #87 #108
- As a user, I can easily navigate the website.

Added by darin3
6 References

Add changes from updated base code
#110 opened by darin3
Task
2 linked pull requests

Used fixed ports (8000/3000)
#76 opened by Kmbear3
Task
1 linked pull request

Fix/ update dependencies
#75 opened by Kmbear3
Task
1 linked pull request

Automated as Done — Manage

17 Closed PRs

Update sprint.md
#45 opened by bonnie-123
Planning
Changes requested

Lani bio/pic
#44 opened by bonnie-123
Planning
Changes approved

Add Bio Header
#48 opened by honeybadgerofdoom
Planning
Changes approved

Lani- update table
#42 opened by bonnie-123
Planning
Changes approved

Add kass bio and pic
#40 opened by Kmbear3
Planning

Automated as Done — Manag

Takeaways and Review

- Plan thoroughly from the start instead of jumping into implementation.
    - Faster development
    - Better product
- Make general designs, then add details later.
    - During implementation, ideas and structure will change.
- Don’t make assumptions on requirements.
    - Always verify misunderstandings with the product owner early on.