Wishify

Progress Report 2

COSC 4P02 Group Project

23rd March 2025

Group Number: 12
Geoffrey Jensen – 7148710
Nicholas Parise – 7242530
Ethan Brennan – 6881411
Stephen Stefanidis – 7140030
Justin Thomas Bijoy – 7123550
Anthony Medico – 3383775

Introduction

This progress report describes the creation of the Wishlist management platform - **Wishify**. Throughout the duration of three development sprints, it offers an in-depth review of the project's progress, focusing on tasks that have been completed, current work, and future goals. The frontend and backend solution, including capabilities for profile customization, Wishlist and event management, and user authentication, are covered in the report. It also keeps track of team contributions, development tools, and meeting summaries to track teamwork and progress.

Sprints

Sprint 1 – **Completed**

1. Design UI for all pages – **Completed**
2. Finish database UML – **Completed**
3. Setup PostgreSQL database – **Completed**
4. Complete placeholder of Wishlist frontend – **Completed**
5. Complete placeholder of sign up and sign in frontend – **Completed**
6. Complete placeholder of list of Wishlist – **Completed**
7. Create a Landing Page – **Completed**
8. Complete placeholder profile page – **Completed**
9. Complete placeholder of event frontend – **Pushed to Sprint 2**
10. Complete placeholder of list of events – **Pushed to Sprint 2**

Sprint 2 – **Completed**

1. Setup production environment – **Completed**
2. Implement frontend for sign in pages – **Completed**
3. Complete placeholder of list of events – **Completed**
4. Authentication API Endpoints – **Completed**
5. Research Pop Up Library – **Completed**
6. Dummy Items for Database – **Completed**
7. Wishlist API Endpoints – **Completed**
8. Event API endpoints – **Completed**
9. Design API specification – **Pushed to Sprint 3**
10. User / I want to create and modify accounts – **Pushed to Sprint 3**
11. Implement frontend for home page – **Pushed to Sprint 3**
12. Complete placeholder of event frontend – **Pushed to Sprint 3**
13. CSS/Styling of Pages – **Pushed to Sprint 3**
14. Express API Testing Framework – **Pushed to Sprint 3**

Sprint 3 – **Completed**

1. Complete placeholder of event frontend – **Completed**
2. User / I want to delete Wishlist – **Completed**

3. Party Host / I want to set deadlines on Wishlist – **Completed**
4. Research OAuth / Firebase - **Completed**
5. Wishlist Contributor / I want to be able to sort a list by price, priority and contribution status. - **Completed**
6. List of Prefix words for profile interests/ and implement to frontend - **Completed**
7. Navigation Bar for all pages – **Completed**
8. Event Planner / I want to duplicate Wishlist – **In progress – Pushed to Sprint 4**
9. Wishlist Contributor / I want a blind feature – **In progress – Pushed to Sprint 4**
10. Design Favicon Logo – **In progress – Pushed to Sprint 4**
11. Implement frontend for home page – **Pushed to Sprint 4**
12. User / I want to create Wishlist – **In progress – Pushed to Sprint 4**
13. CSS/Styling of Pages – **In progress – Pushed to Sprint 4**
14. Create and modify user accounts – **In progress – Pushed to Sprint 4**

Sprint 4 – **Completed**

1. Implement frontend for home page – **Completed**
2. User / I want to delete Wishlist – **Completed**
3. API Contributions – **Completed**
4. User / I want to set interests on my profile that other people can see – **Completed**
5. API Categories – **Completed**
6. Implement frontend of event pages – **Completed**
7. Write Pytest for end-to-end testing – **Completed**
8. API add support for arrays – **Completed**
9. API Status page – **Completed**
10. Event Planner / I want to duplicate Wishlist – **Pushed to Sprint 5**
11. Wishlist Contributor / I want a blind feature – **In progress - Pushed to Sprint 5**
12. User / I want to create and modify accounts – **In progress - Pushed to Sprint 5**
13. User / I want to create Wishlist – **Pushed to Sprint 5**
14. CSS styling of Pages – **In progress - Pushed to Sprint 5**
15. Implement Wishlist share feature – API – **In progress - Pushed to Sprint 5**
16. Implement email Notifications – **In progress - Pushed to Sprint 5**
17. Wishlist Owner / I want to share my Wishlist with a link – frontend – **In progress - Pushed to Sprint 5**
18. User / I want to be able to opt – in for notifications on some of my lists to see when the status of an item is changed – **Pushed to Sprint 5**
19. User / I want to be able to leave a comment on contributions – **In progress - Pushed to Sprint 5**
20. Wishlist Owner / I want a summary page that shows contributions at a glance – **Pushed to Sprint 5**

Sprint 5 – **In Progress**

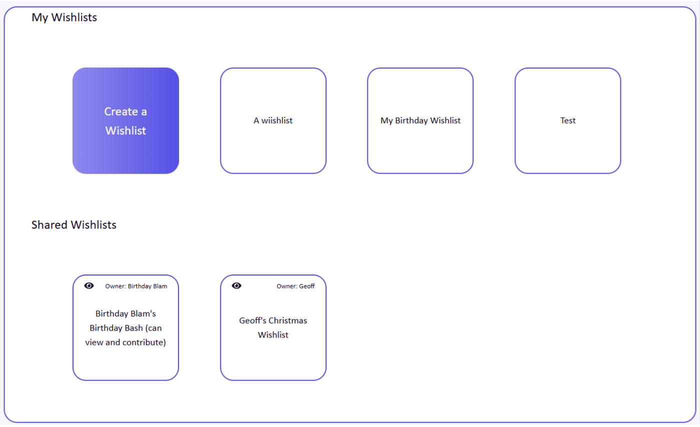
1. Event Planner / I want to duplicate Wishlist – **In progress**
2. Implement Wishlist share feature – API – **Completed**
3. CSS Styling of Pages – **Pushed to Backlog**

4. Implement email Notifications – **In progress**
5. Wishlist Owner / I want to share my Wishlist with a link – frontend – **In progress**
6. Wishlist Contributor / I want a blind feature – **In progress**
7. User / I want to be able to leave a comment on contributions – **Completed**
8. Event Planner / I want to duplicate wishlists - **Completed**
9. User / I want to create and modify accounts – **In progress**
10. User / I want to create Wishlist – **In progress**
11. User / I want to be able to opt-in for notifications on some of my lists to see when the status of an item is changed
12. Wishlist Owner / I want a summary page that shows contributions at a glance - **In progress**
13. Connect homepage to backend - **Completed**
14. Add notification for homepage - **In progress**
15. Add tests for contributions
16. Add tests for items
17. Add tests for categories
18. Creation of Help web page
19. Connect events page to backend
20. Layout and styling of email notification
21. Implement notification inbox Backend - **Completed**
22. Site Owner/ I would like most wishlisted items to be tracked
23. Forget password - backend
24. Frontend Sharing Wishlists - page

Frontend Progress

CSS Styling

The last few weeks we've made progress on creating a uniform style for our web app and applying it to our pages.



My Wishlists

Create a Wishlist

A wiishlist

My Birthday Wishlist

Test

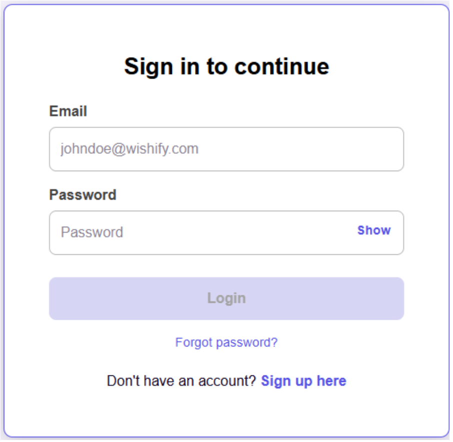
Shared Wishlists

Owner: Birthday Blam

Birthday Blam's Birthday Bash (can view and contribute)

Owner: Geoff

Geoff's Christmas Wishlist



Sign in to continue

Email

johndoe@wishify.com

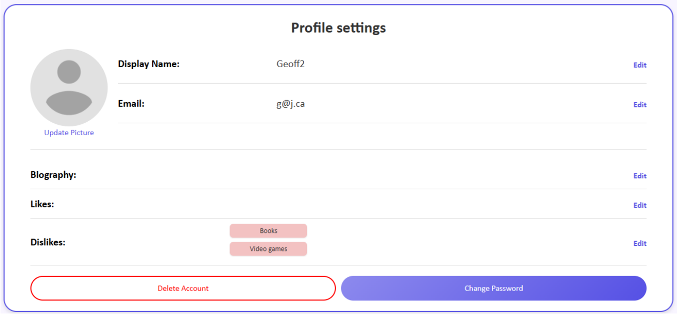
Password

Password Show

Login

Forgot password?

Don't have an account? Sign up here



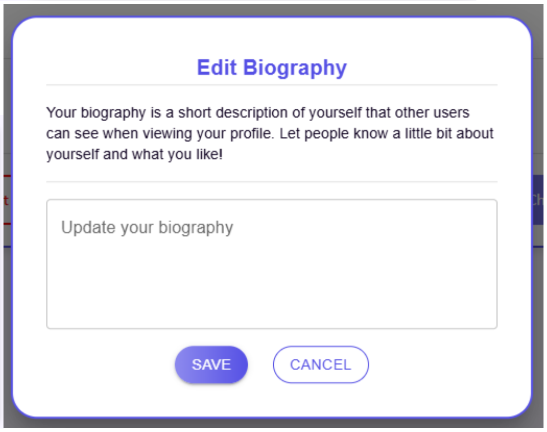
Profile settings

Update Picture

Display Name:	Geoff2	Edit
Email:	g@j.ca	Edit
Biography:		Edit
Likes:		Edit
Dislikes:	Books Video games	Edit

Delete Account

Change Password



Edit Biography

Your biography is a short description of yourself that other users can see when viewing your profile. Let people know a little bit about yourself and what you like!

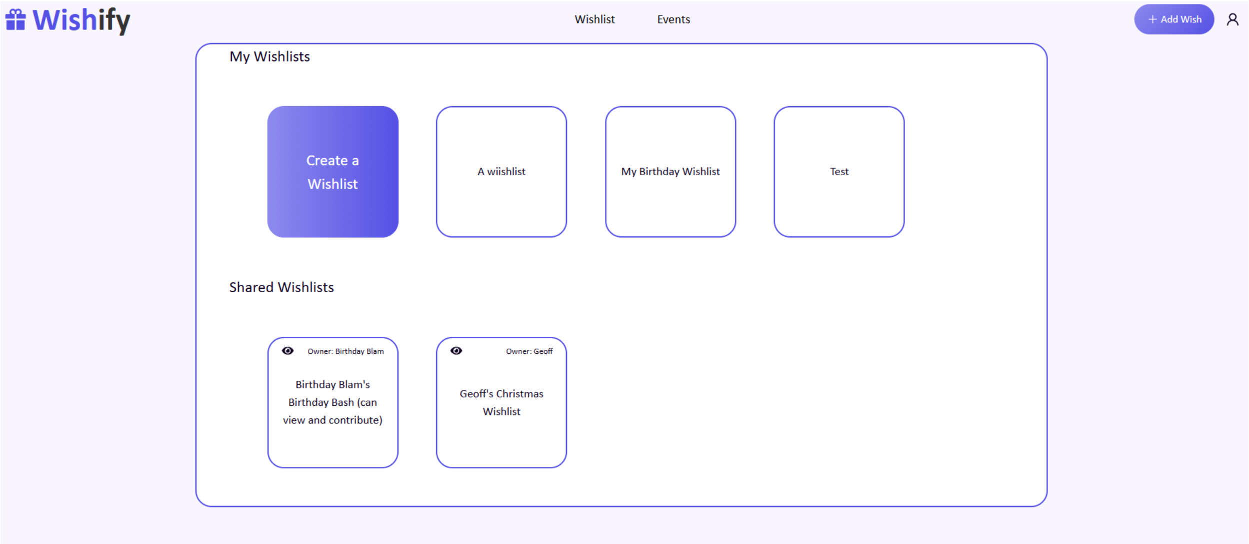
Update your biography

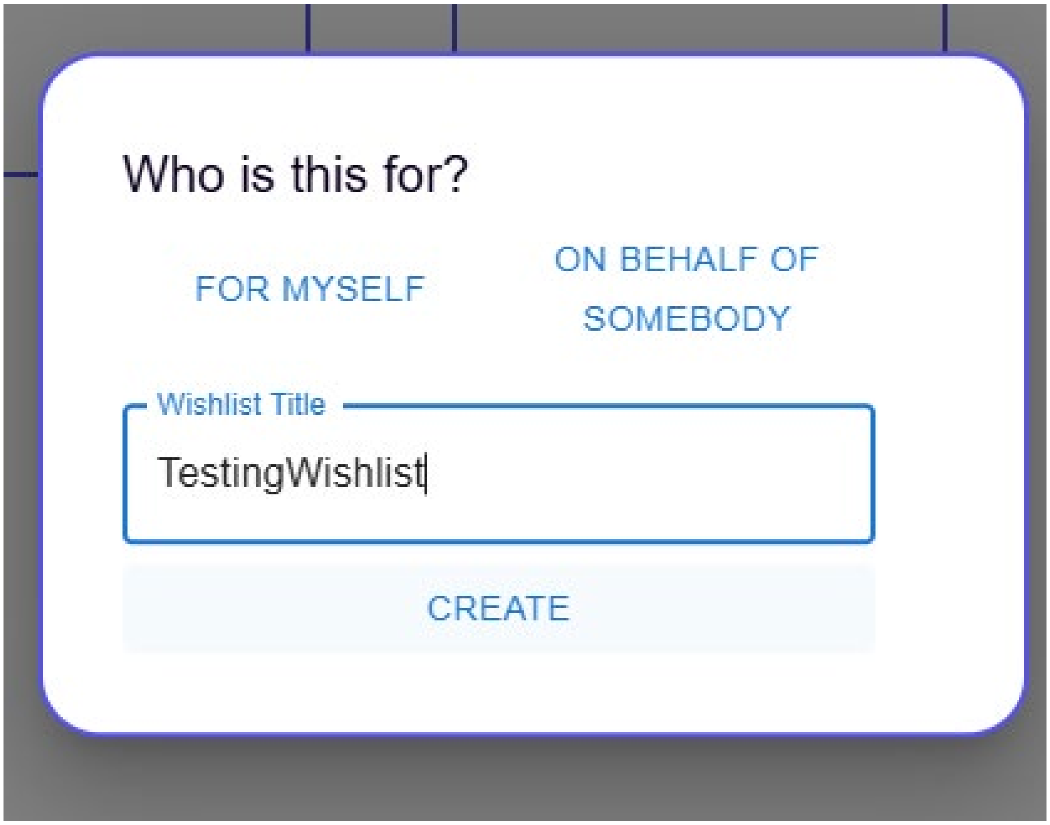
SAVE

CANCEL

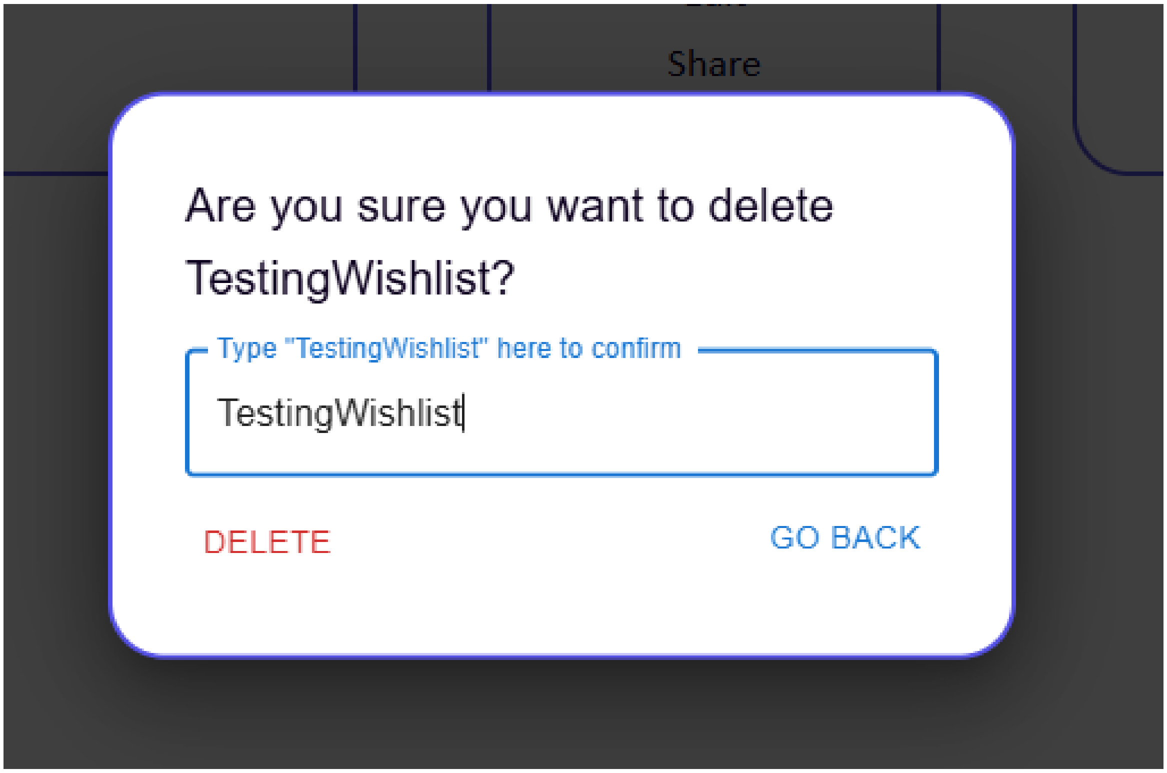
Wishlist Home Page

The Wishlist section is where you can view and manage all your saved Wishlist in one place. Here, you'll have full control over your lists with the ability to add, edit, and delete Wishlist as needed. The CSS has been updated and the ability to rename, delete, duplicate and save wishlists to the backend have all been added since the last report.



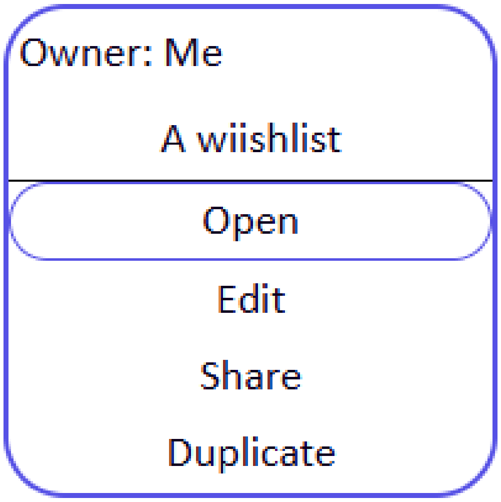


Wishlist Title: TestingWishlist



Are you sure you want to delete TestingWishlist?

Type "TestingWishlist" here to confirm: TestingWishlist

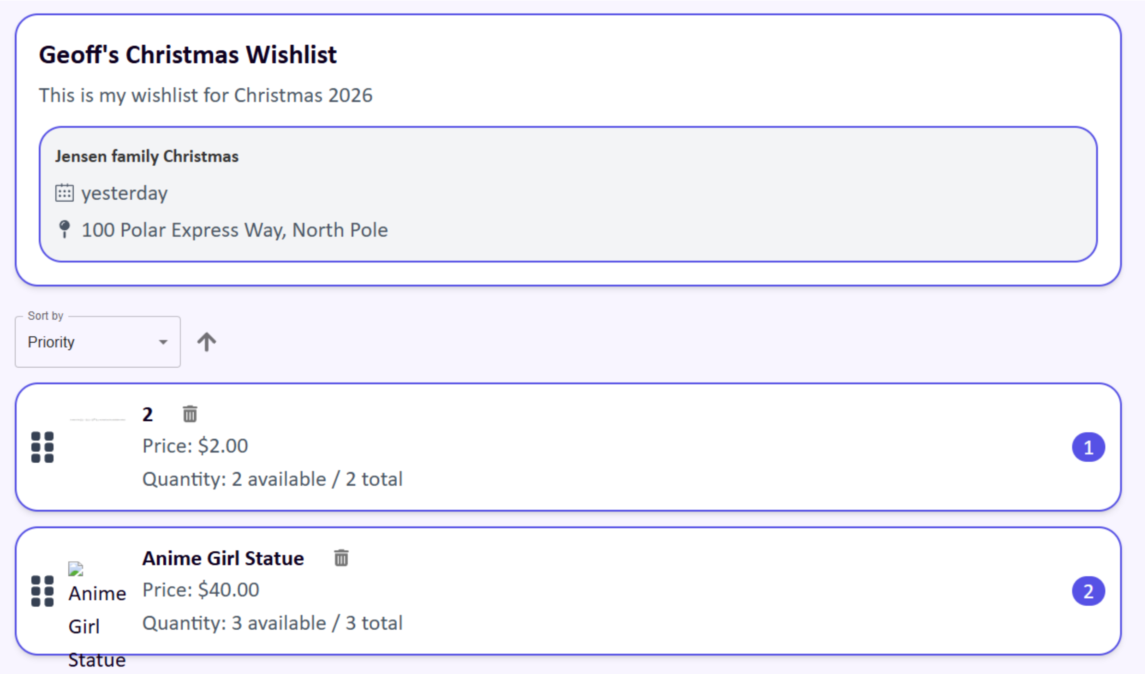


Owner: Me

A wiishlist

Wishlist Page

When you enter a Wishlist, you'll be redirected to the Wishlist Details page, where you can fully manage its contents. Since the last report, deleting items and posting reservations/contributions to items have been implemented.



Geoff's Christmas Wishlist

This is my wishlist for Christmas 2026

Jensen family Christmas

yesterday

100 Polar Express Way, North Pole

Sort by: Priority ↑

2

Price: $2.00

Quantity: 2 available / 2 total

1

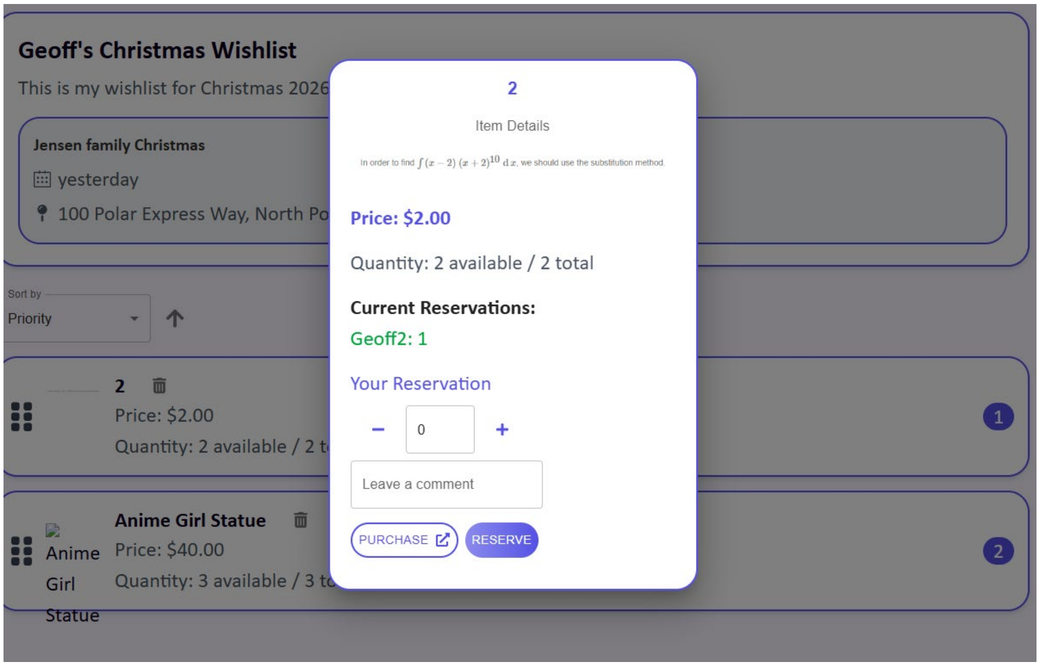
Anime Girl Statue

Anime Girl Statue

Price: $40.00

Quantity: 3 available / 3 total

2



2

Item Details

In order to find $\int (x-2)\,(x+2)^{10}\,\mathrm{d}\,x$, we should use the substitution method.

Price: $2.00

Quantity: 2 available / 2 total

Current Reservations:

Geoff2: 1

Your Reservation

− 0 +

Leave a comment

Geoff's Christmas Wishlist

This is my wishlist for Christmas 2026

Jensen family Christmas

yesterday

100 Polar Express Way, North Po

Sort by: Priority ↑

2

Price: $2.00

Quantity: 2 available / 2 t

1

Anime Girl Statue

Anime Girl Statue

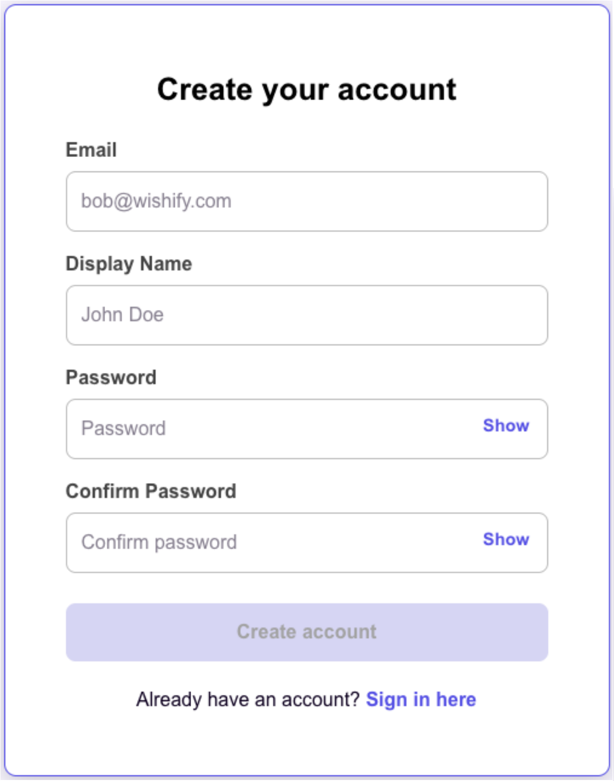
Price: $40.00

Quantity: 3 available / 3 to

2

Sign-Up and Login Pages

The sign up and login pages have been complete and fully functional since the last progress report, however they have now been refreshed with CSS changes to match the overall site aesthetic.



Create your account

Email

bob@wishify.com

Display Name

John Doe

Password

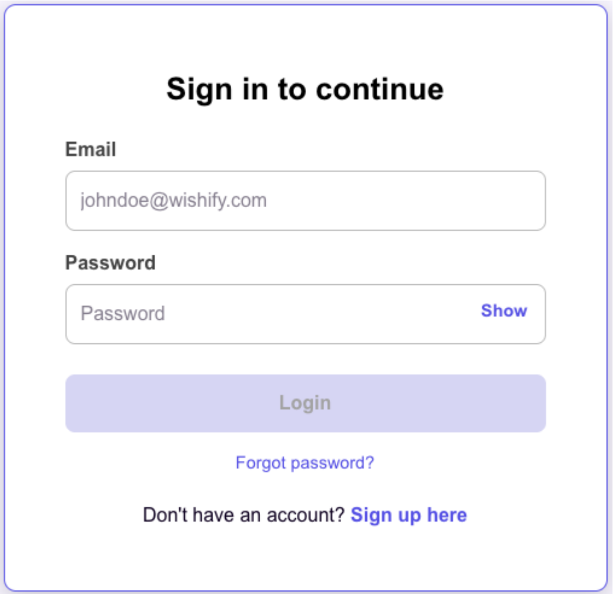
Password Show

Confirm Password

Confirm password Show

Create account

Already have an account? Sign in here



Sign in to continue

Email

johndoe@wishify.com

Password

Password Show

Login

Forgot password?

Don't have an account? Sign up here

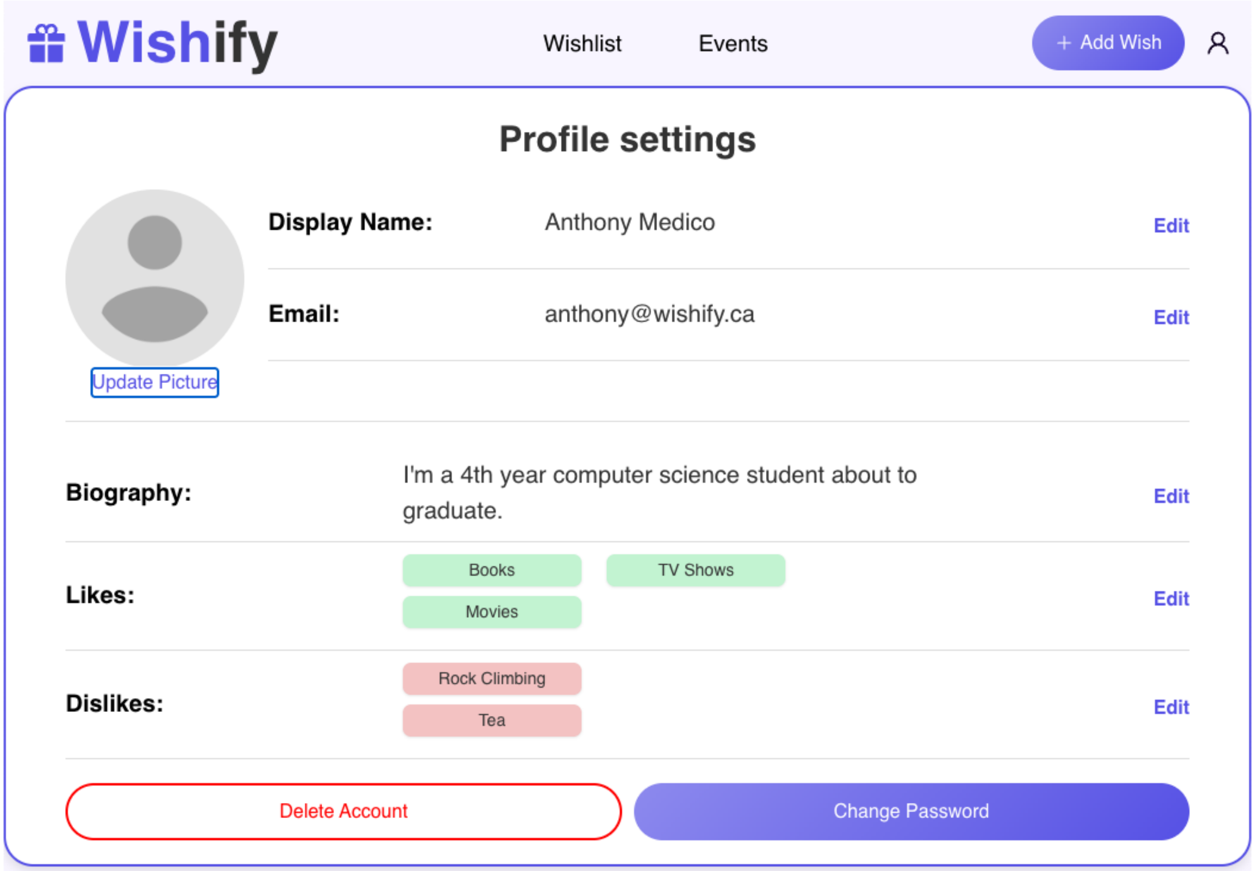
Future improvements: the team is still considering the use of OAuth and logging in via social media credentials. This will be implemented if time permits.

Profile Settings Page

The profile settings page is where the user can update various account settings as well as personal information. They may edit their profile picture, display name, biography, likes and dislikes, email and password. Here the user can also delete their account.

Likes and dislikes is a predetermined list of categories stored on the backend that the user can select from to include in their profile. This lets other users know what categories of gifts would be best for them, as well as categories to avoid.

This page has also been refreshed with CSS changes to match the overall aesthetic of the site. An overview is shown below.



Profile settings

Display Name: Anthony Medico

Email: anthony@wishify.ca

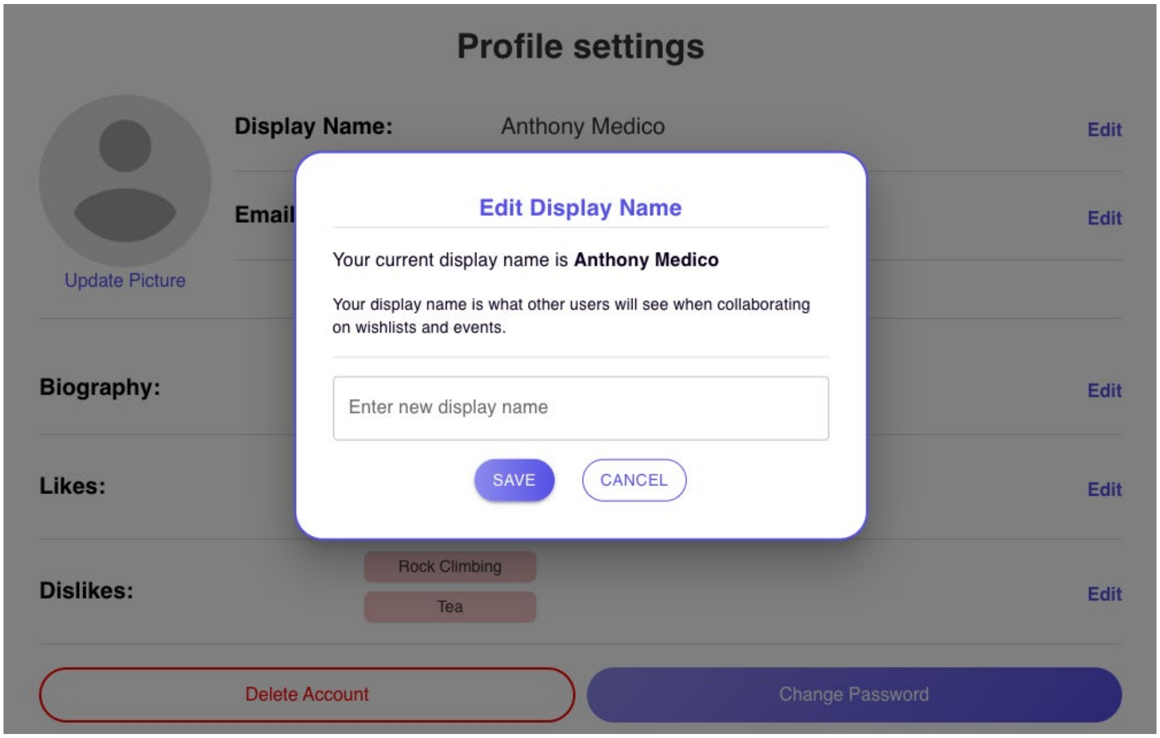
Biography: I'm a 4th year computer science student about to graduate.

Likes: Books, TV Shows, Movies

Dislikes: Rock Climbing, Tea

Only the frontend UI was implemented as of last progress report, and since then all of the logic to edit fields, and handle adding and removing of like/dislike categories has been implemented. It has also been connected to the backend so changes persist. Aside from a few bugs to be fixed before the end of this sprint, it is fully functional. To handle edits, custom modals were created for each field which will display upon the user clicking its corresponding edit button.

The two screenshots below show the modals to edit your display name and biography. We ensured the existing biography text appears in the textbox for easy edits.


Profile settings

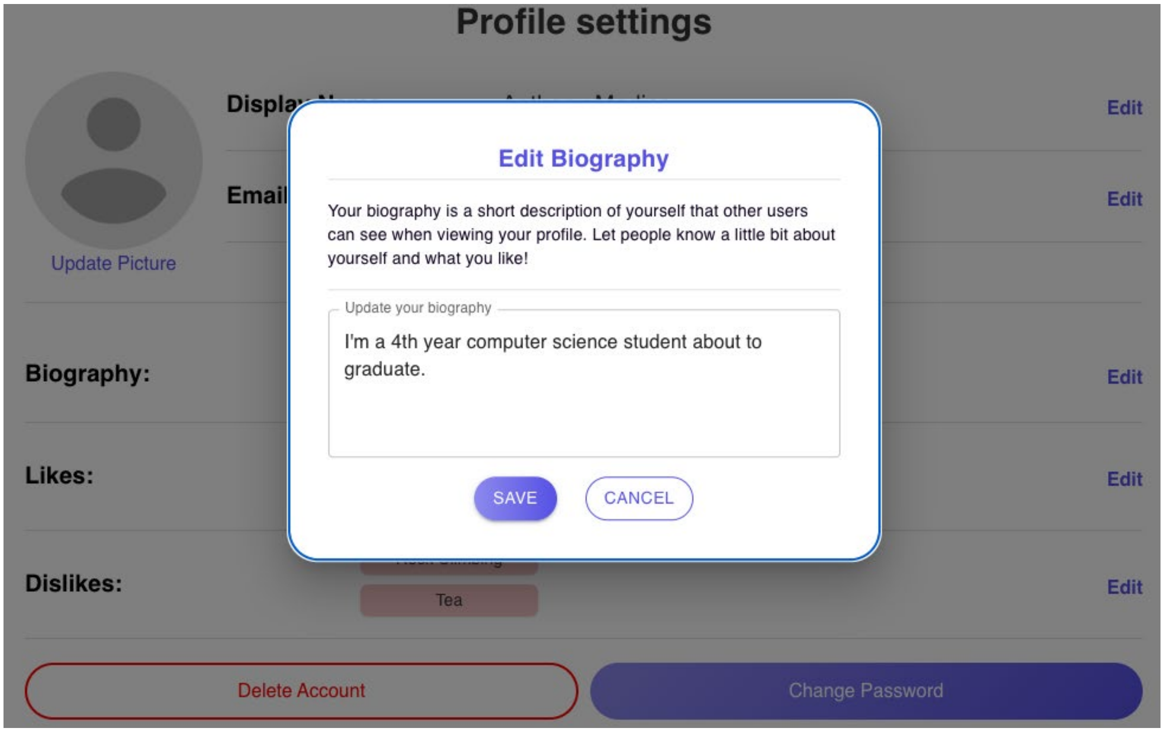
Display Name: Anthony Medico

Edit Display Name

Your current display name is **Anthony Medico**

Your display name is what other users will see when collaborating on wishlists and events.

Enter new display name


Profile settings

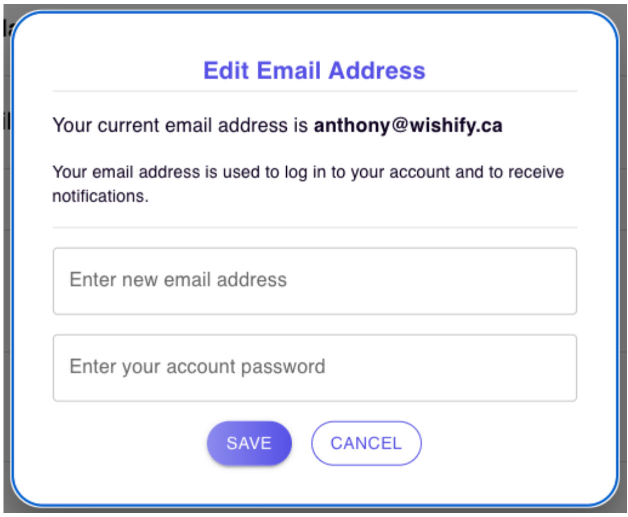
Edit Biography

Your biography is a short description of yourself that other users can see when viewing your profile. Let people know a little bit about yourself and what you like!

Update your biography

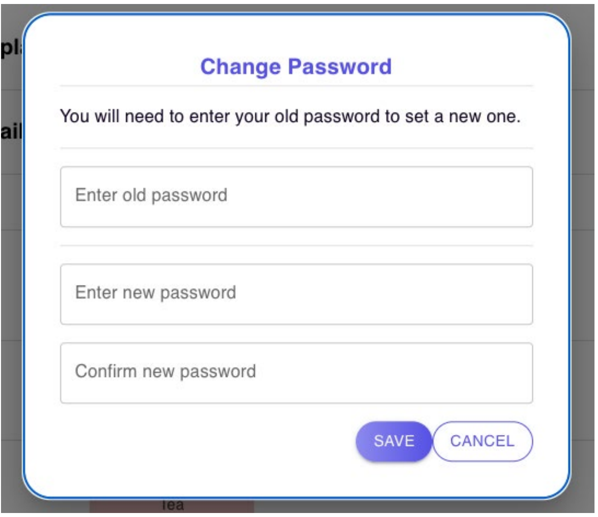
I'm a 4th year computer science student about to graduate.

The edit email and change password modals are shown below. These require the input of the user's existing password to change.


Edit Email Address

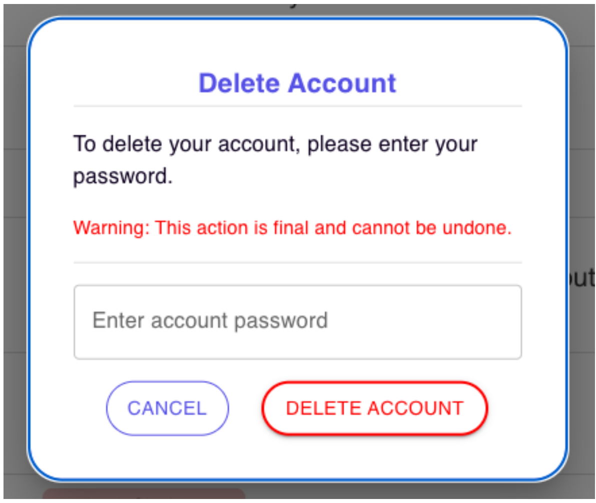
Your current email address is **anthony@wishify.ca**

Your email address is used to log in to your account and to receive notifications.


Change Password

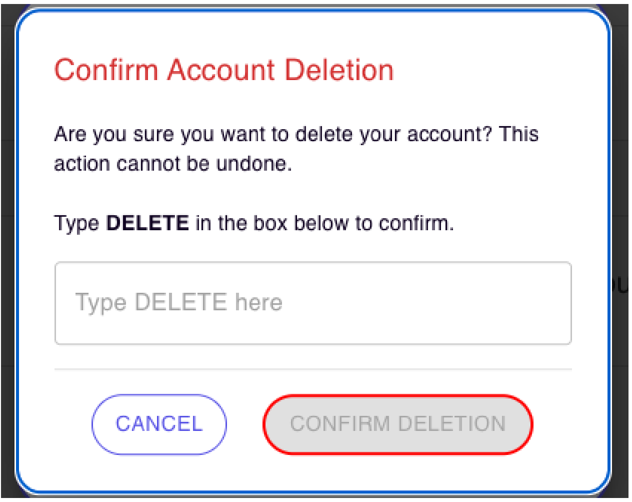
You will need to enter your old password to set a new one.

Deleting an account is a 2-step process which requires the user's password, as well as an explicit confirmation. The two modals to handle this are shown below.


Delete Account

To delete your account, please enter your password.

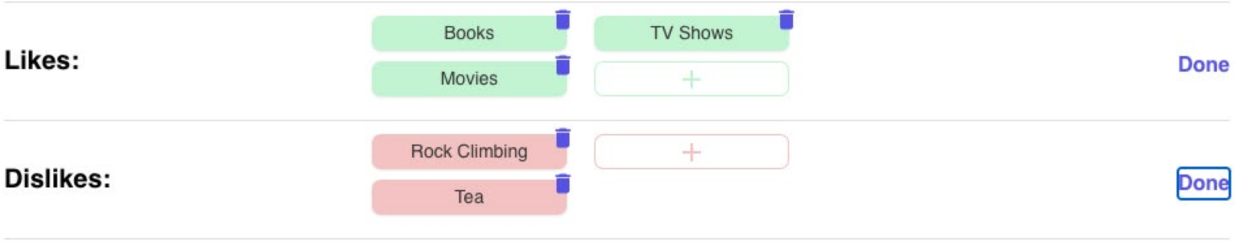
Warning: This action is final and cannot be undone.


Confirm Account Deletion

Are you sure you want to delete your account? This action cannot be undone.

Type **DELETE** in the box below to confirm.

Lastly, to add and remove likes/dislikes, the user will click the corresponding edit button to enable edit mode. From here they can click the trash icon to delete specific categories, or click the + icon to display a modal where they can select from the list of categories.



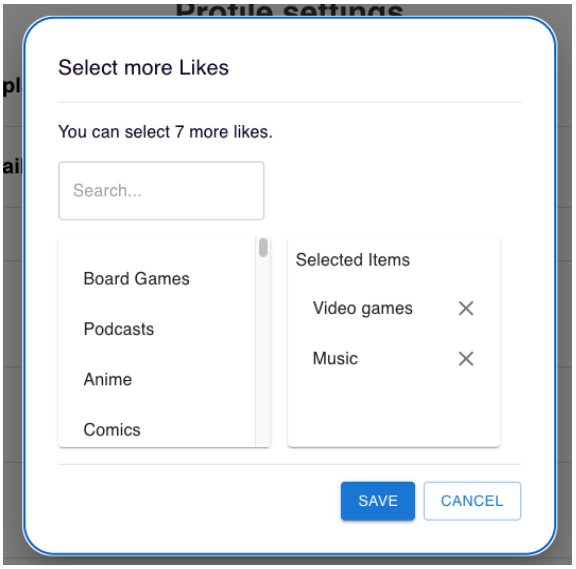
Likes: Books, TV Shows, Movies, +

Done

Dislikes: Rock Climbing, +, Tea

Done

When selecting a new category, the modal will automatically fetch the list of categories from the server, and only display those that are not currently one of the user's likes or dislikes. The modal allows the user to select multiple and then save them all in one go. In the left screenshot below, the modal displays the list of available categories on the left, and the newly selected ones on the right (which can be unselected by clicking the X. Once the new ones have been selected, the user can click save which will save the new settings to the backend and update the frontend display. There is also a search bar where users can begin typing a category name which will filter the list below. The right screenshot shows an example where the user begins typing the first few letters for "video games" and the list filters to show it.



Profile settings

Select more Likes

You can select 7 more likes.

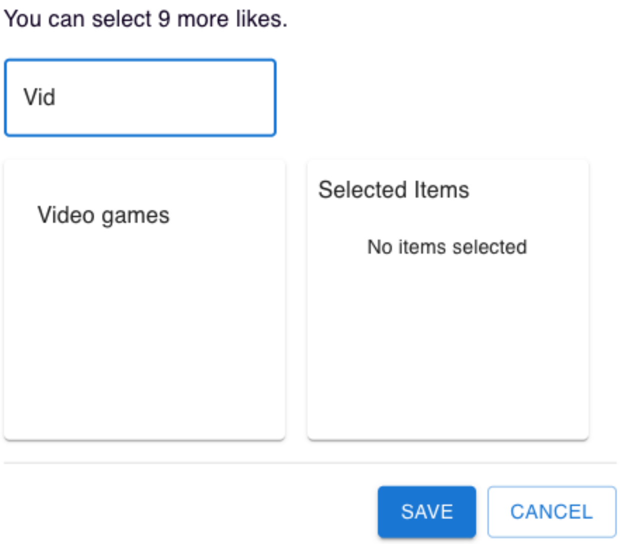
Search...

Board Games
Podcasts
Anime
Comics

Selected Items

Video games ✕
Music ✕

SAVE CANCEL



You can select 9 more likes.

Vid

Video games

Selected Items

No items selected

SAVE CANCEL

Backend Progress

Backend had a lot of important features added in these last sprints. Some of the most important features are contributions, categories, notifications and email support. Along with many quality of life improvements to make interacting with the API much easier for the front end team. The progress on the backend is much more than the progress on the front end, in fact in terms of the features we intended to deliver from our release planning document, the backend is almost completed. Following this point most backend features are now new features to make the product even better.

Due to the criticality of the backend in terms of its requirements for correctness and uptime, the API cannot fail in production. Therefore the backend is following a Incremental development model, where a subset of required endpoints are added at a time, tested and then finally deployed. In the event of bugs or new requirements needed for the front end the same process is followed, where that end point is changed then tested and finally redeployed.

To make deployment easier we use github actions to automatically deploy pushes to our server. This means that deploying new features is very easy and takes no effort. The downside is that means that errors can be pushed to the server causing issues. In the last month we have encountered several issues where typescript errors have prevented a frontend build, there is no way to fix these errors unfortunately and just require our team to be more diligent before pushing their files and have better review processes before merging branches.

API additions

users:	
POST /users/categories/:categoryId	Assign a category to logged in user
POST /users/categories	Assign an array of categories to logged in user
Put /users/categories/:categoryId	updates a users love or hate value
Put /users/categories	updates an array of categories to logged in user
DELETE /users/categories/:categoryId	Remove a category from logged in user
DELETE /users/categories	Remove an array of categories from logged in user

Categories:	
GET /categories	Get all categories
GET /categories/:id	Get category details
POST /categories	Create a new category

Wishlists	
POST /wishlists/members	make logged in user a member of the wishlist given the share_token
POST /wishlists/:id/duplicate	Duplicate the wishlist
GET /wishlists/:id/items	Get all items in a wishlist (and contributions)
GET /wishlists/share/:token	get the shared wishlist
POST /wishlists/share	share the wishlist to email, if user exists membership added. else send email

Contributions:	
GET /contributions	Get all contributions from logged in user
GET /contributions/wishlists/:id	Get all contributions from wishlist
GET /contributions/items/:id	Get contributions for an item
POST /contributions	Add a contribution (given item_id)
PUT /contributions/:id	Update a contribution (mark as purchased, etc.)
DELETE /contributions/:id	Remove a contribution

Notifications:	
GET /notifications	Get all notifications from logged in user
PUT /notifications/:id	Edit a notifications (ex: is_read)
DELETE /notifications/:id	Delete a notifications

Pending Tasks

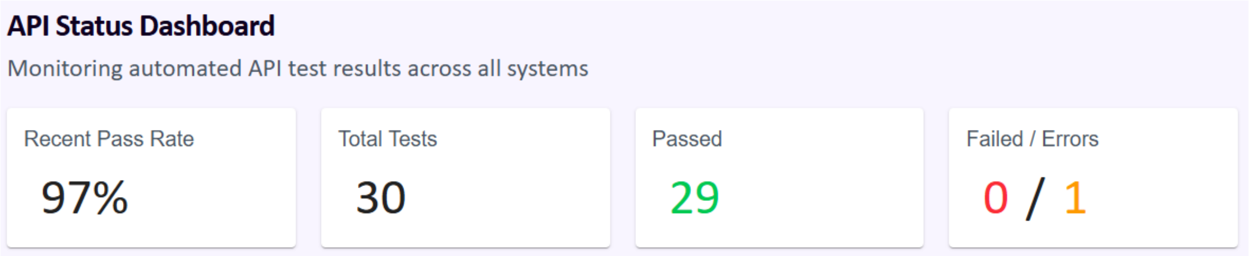
- **Extending notification and email support:** currently only a few select actions will send notifications and/or emails, this feature needs to be used more
- **payment support:** for pro features we need to accept payment to give users pro status
- **Data tracking:** Simple data tracking is planned, we want to track users' most requested items and most liked and disliked categories.

Testing

Testing has made significant progress in these last few sprints, we have now gone from completely manual testing to automated testing with a status page visible to all users. To do this we have set up an automated testing flow on our server, which feeds scheduled testing results to our API Status page (https://www.wishify.ca/status).

- Our tests are for testing the end-to-end behaviour and interface of our backend API
- We are using pytest as our testing framework, with the json-report plugin, and the python requests library for facilitating HTTP requests.
- Our automated tests are currently configured to run every 30 minutes, on the hour (XX:00) and half hour (XX:30).

Our status page provides the summary statistics of the most recent test

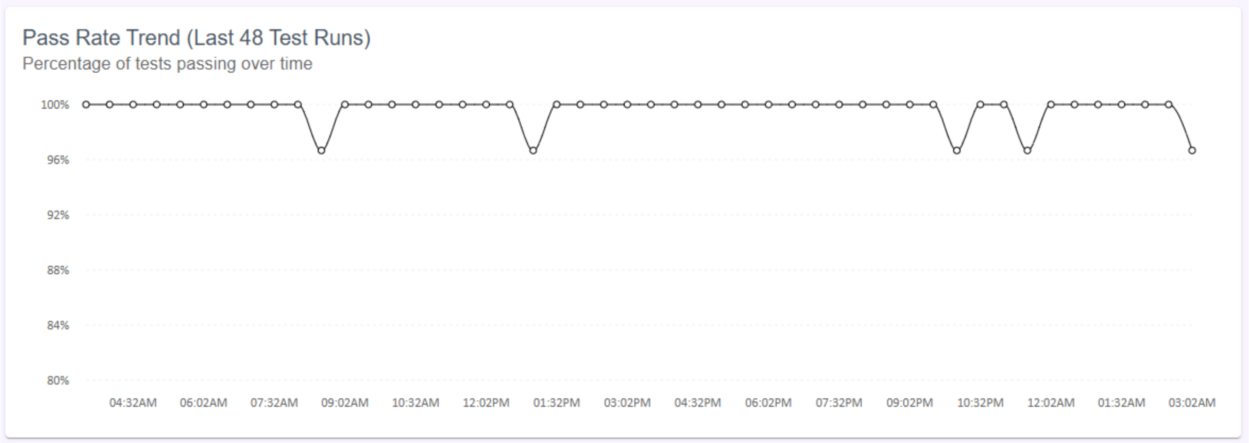


API Status Dashboard

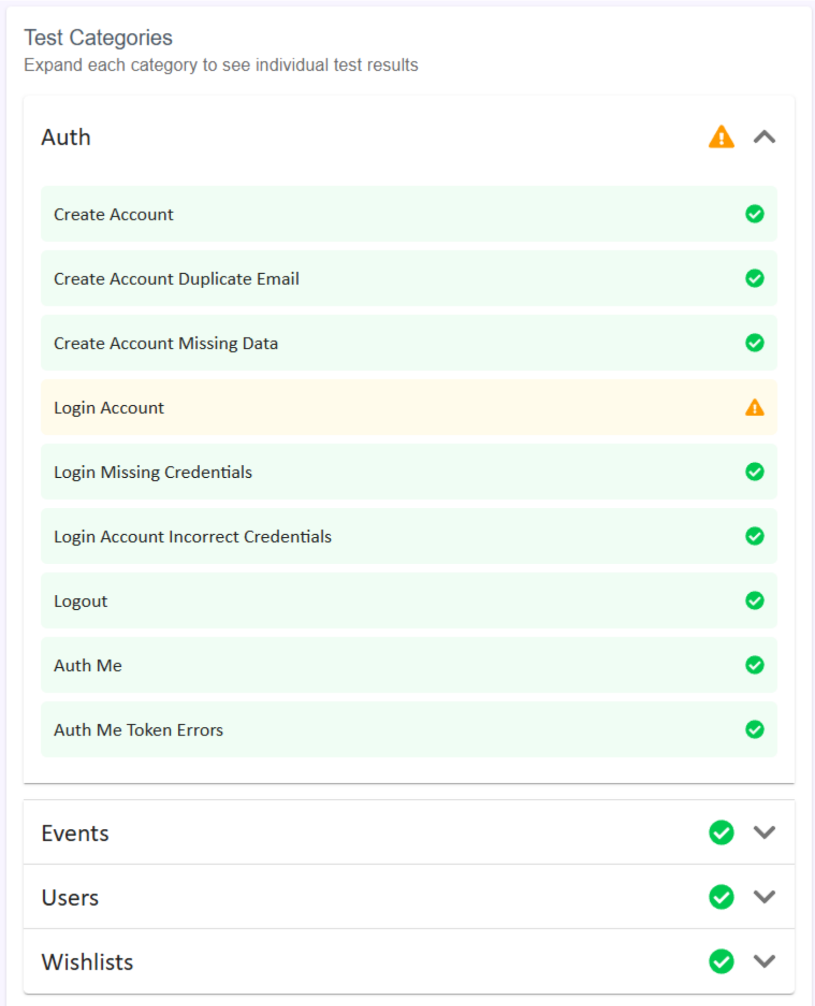
Monitoring automated API test results across all systems

Recent Pass Rate	Total Tests	Passed	Failed / Errors
97%	30	29	0 / 1

This page also shows a history of the last 48 test runs in a nice graph (48 test runs, with 2 tests per hour means we show all the test runs from the last 24 hours).



Also on this screen are the individual results of the latest tests, and you can see the result of every individual test.



Test Categories

Expand each category to see individual test results

Auth ⚠

- Create Account ✓
- Create Account Duplicate Email ✓
- Create Account Missing Data ✓
- Login Account ⚠
- Login Missing Credentials ✓
- Login Account Incorrect Credentials ✓
- Logout ✓
- Auth Me ✓
- Auth Me Token Errors ✓

Events ✓

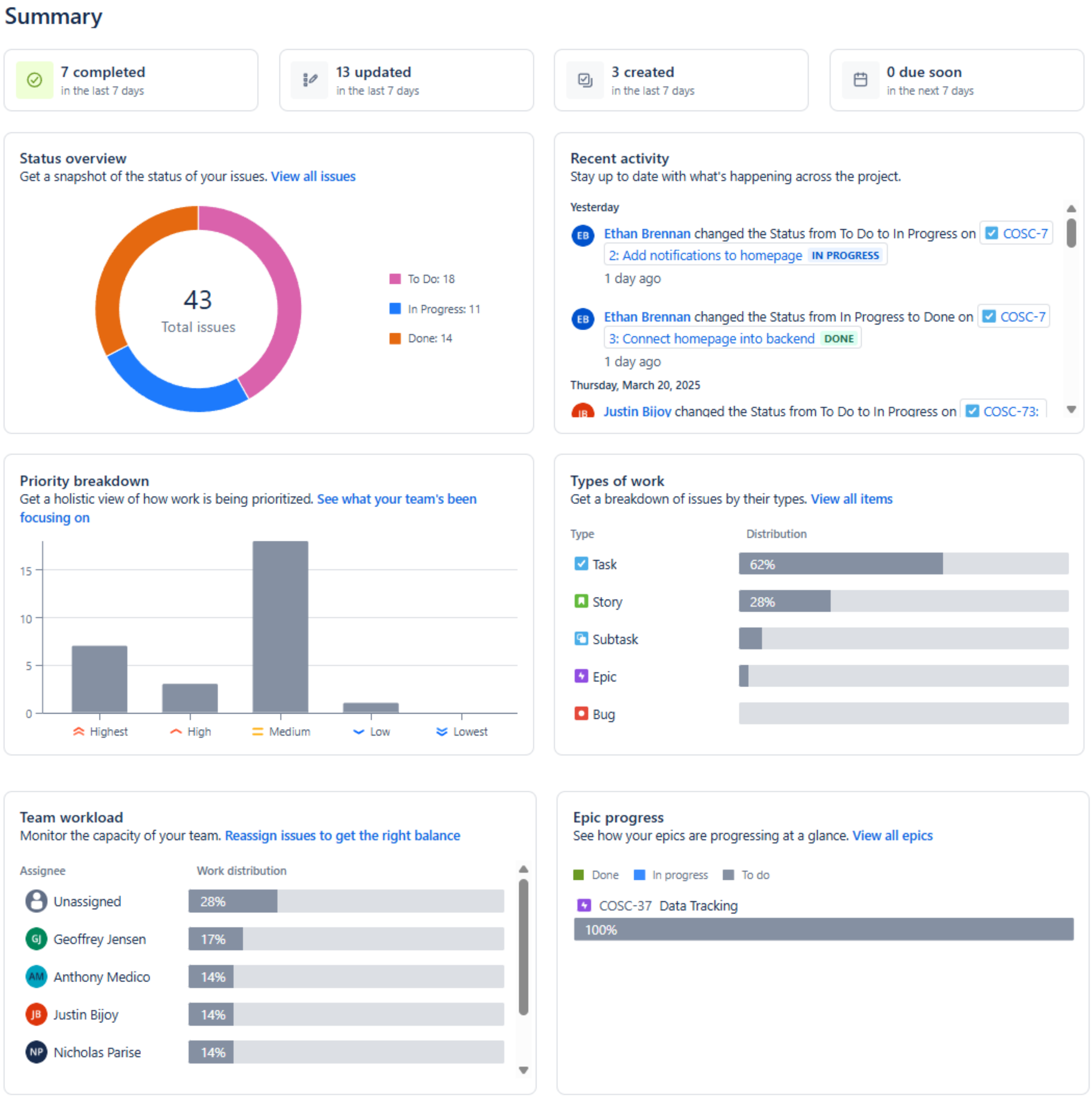
Users ✓

Wishlists ✓

For a much more detailed view on our automated testing flow and how it works, you can look through our documentation.

Tools

Jira



Summary

✓ 7 completed in the last 7 days

13 updated in the last 7 days

3 created in the last 7 days

0 due soon in the next 7 days

Status overview

Get a snapshot of the status of your issues. View all issues

Recent activity

Stay up to date with what's happening across the project.

Yesterday

EB Ethan Brennan changed the Status from To Do to In Progress on ☑ COSC-72: Add notifications to homepage IN PROGRESS

1 day ago

EB Ethan Brennan changed the Status from In Progress to Done on ☑ COSC-73: Connect homepage into backend DONE

1 day ago

Thursday, March 20, 2025

JB Justin Bijoy changed the Status from To Do to In Progress on ☑ COSC-73:

Priority breakdown

Get a holistic view of how work is being prioritized. See what your team's been focusing on

Types of work

Get a breakdown of issues by their types. View all items

Type	Distribution
Task	62%
Story	28%
Subtask	
Epic	
Bug	

Team workload

Monitor the capacity of your team. Reassign issues to get the right balance

Assignee	Work distribution
Unassigned	28%
GJ Geoffrey Jensen	17%
AM Anthony Medico	14%
JB Justin Bijoy	14%
NP Nicholas Parise	14%

Epic progress

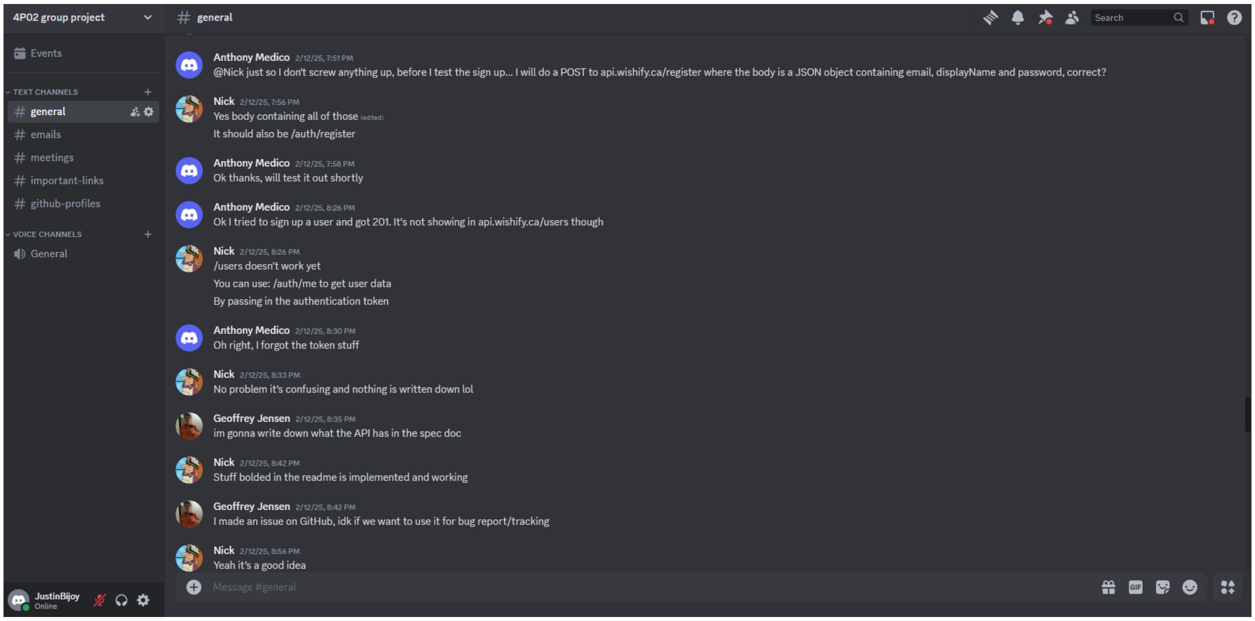
See how your epics are progressing at a glance. View all epics

🟩 Done 🟦 In progress ■ To do

COSC-37 Data Tracking

100%

Discord



general

Anthony Medico 2/12/25, 7:51 PM
@Nick just so I don't screw anything up, before I test the sign up... I will do a POST to api.wishify.ca/register where the body is a JSON object containing email, displayName and password, correct?

Nick 2/12/25, 7:56 PM
Yes body containing all of those (edited)
It should also be /auth/register

Anthony Medico 2/12/25, 7:58 PM
Ok thanks, will test it out shortly

Anthony Medico 2/12/25, 8:26 PM
Ok I tried to sign up a user and got 201. It's not showing in api.wishify.ca/users though

Nick 2/12/25, 8:26 PM
/users doesn't work yet
You can use: /auth/me to get user data
By passing in the authentication token

Anthony Medico 2/12/25, 8:30 PM
Oh right, I forgot the token stuff

Nick 2/12/25, 8:33 PM
No problem it's confusing and nothing is written down lol

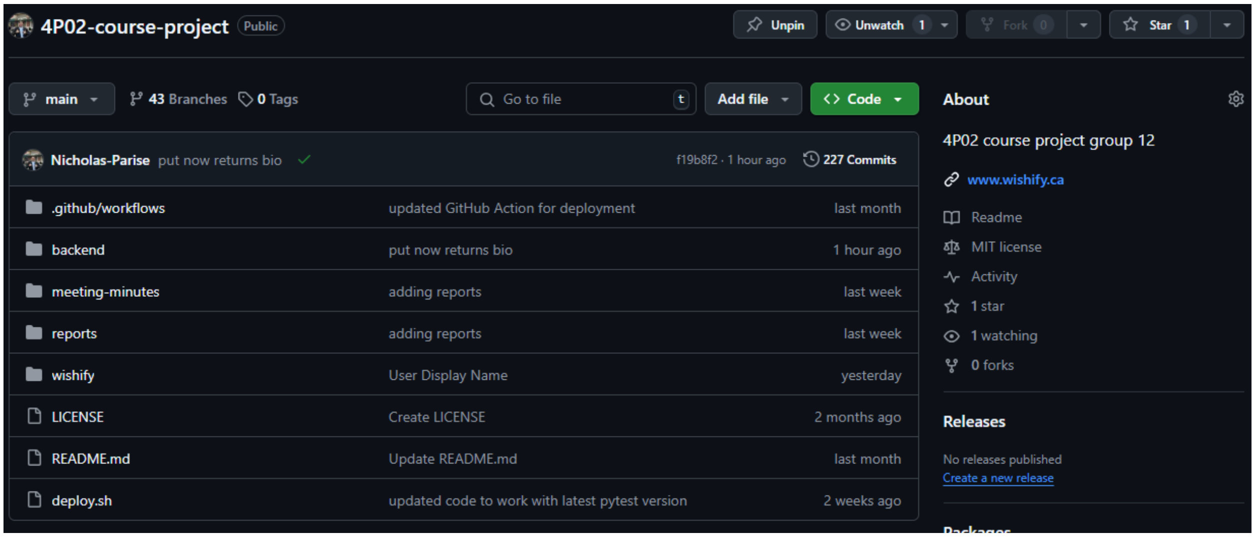
Geoffrey Jensen 2/12/25, 8:35 PM
im gonna write down what the API has in the spec doc

Nick 2/12/25, 8:42 PM
Stuff bolded in the readme is implemented and working

Geoffrey Jensen 2/12/25, 8:42 PM
I made an issue on GitHub, idk if we want to use it for bug report/tracking

Nick 2/12/25, 8:56 PM
Yeah it's a good idea

GitHub



Nicholas-Parise put now returns bio ✓ f19b8f2 · 1 hour ago 227 Commits

.github/workflows	updated GitHub Action for deployment	last month
backend	put now returns bio	1 hour ago
meeting-minutes	adding reports	last week
reports	adding reports	last week
wishify	User Display Name	yesterday
LICENSE	Create LICENSE	2 months ago
README.md	Update README.md	last month
deploy.sh	updated code to work with latest pytest version	2 weeks ago

About

4P02 course project group 12

www.wishify.ca

Readme
MIT license
Activity
1 star
1 watching
0 forks

Releases

No releases published
Create a new release

Meetings

Meeting: Sprint Retrospect & Kick off Sprint 4

- **Date:** 28th February 2025
- **Time:** 11:00 AM – 2:00 PM
- **Venue:** Library Room STH503C
- **Key Points:**
  - **Sprint Retrospect:** Reviewed Sprint 3 progress, discussing achievements and areas for improvement. All outstanding items from Sprint 3 were pushed to Sprint 4.
  - **Jira Update:** Updated the Jira board with completed and pending tasks. Ensured all tasks are assigned correctly and in the right status for Sprint 4.
  - **Review Outstanding Items:** Wishlist home page placeholder was still outstanding; users need to be redirected after successful login. CSS design needs implementation, and research on features, testing, and email notification is required.
  - **Kick off Sprint 4:** Defined objectives and priorities for Sprint 4. Assigned new tasks focusing on enhancing the Wishlist and events features.
- **Next Steps:** Team members to begin implementation of assigned tasks. Finalize and document API specifications. Schedule weekly progress review meetings. The next stand-up meeting is on 7th March 2025.

Meeting: Weekly Meeting for Sprint 4

- **Date:** 7th March 2025
- **Time:** 11:00 AM – 2:00 PM
- **Venue:** Library Room STH503A
- **Key Points:**
  - **Review of Completed Work:** Team reviewed completed tasks, providing updates on respective areas to ensure alignment on project milestones.
  - **CSS Update:** Justin demonstrated CSS updates and design implementation. Limited progress noted due to outdated GitHub repository; all branches must be published promptly.
  - **Pytest Framework:** Geoff explained the pytest framework for implementation, finalizing the approach for report management.
  - **Shared Link Feature:** Nick led a discussion on Wishlist link sharing, resulting in a finalized method.
  - **Profile Page Update:** Anthony presented significant updates to the profile page, which were well received.
- **Next Steps:** Team members to continue working on assigned tasks for the sprint's end next week.

Meeting: Sprint Retrospect & Kick off Sprint 5

- **Date:** 14th March 2025
- **Time:** 11:00 AM – 2:00 PM
- **Venue:** Library Room STH503C
- **Key Points:**
    - **Sprint Retrospect:** Reviewed Sprint 4 progress, discussing achievements and areas for improvement. Outstanding items from Sprint 3 were pushed to Sprint 5.
    - **Jira Update:** Updated the Jira board with completed and pending tasks for Sprint 4.
    - **Review Outstanding Items:** CSS styling is incomplete; additional testing is required for Pytest. Research on notification implementation is complete but still needs development. The summary page has been deferred to the next sprint.
    - **Kick off Sprint 5:** Defined objectives and assigned new tasks based on project requirements. Focus on enhancing Wishlist and event features.
- **Next Steps:** Team members to begin implementation of Sprint 5 tasks. The next stand-up meeting is scheduled for 21st March 2025.

Meeting: Progress Check-in with TA

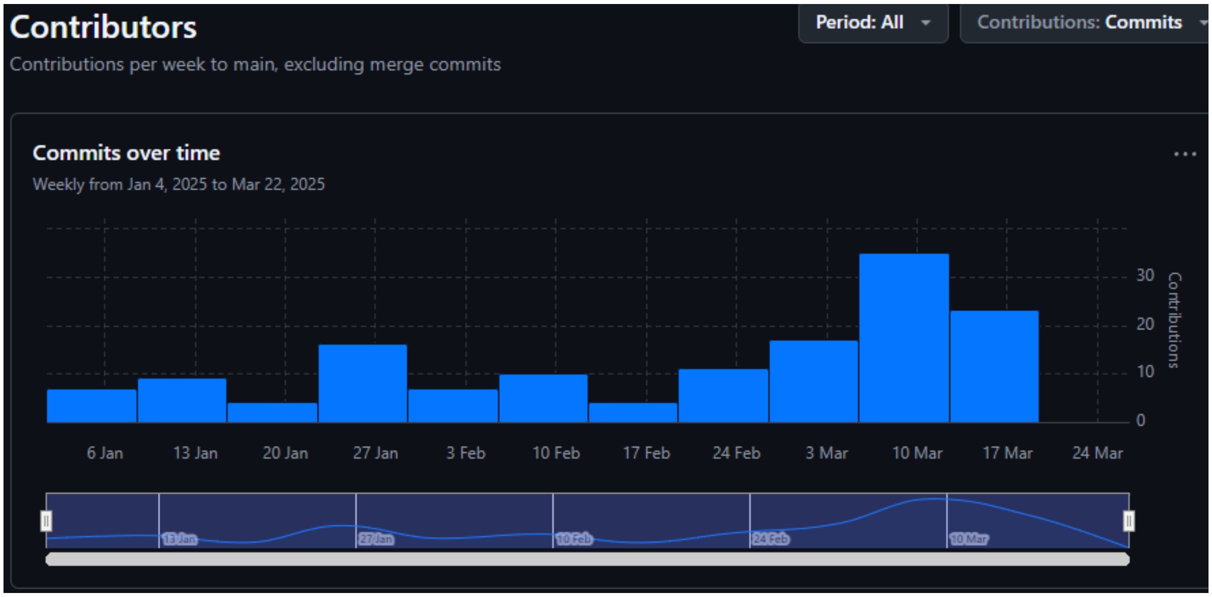
- Date: 18th February 2025
- Time: 9:00 AM – 9:30 AM
- Venue: Online (Microsoft Teams)
- Key Points:
    - **Progress Report:** Presented progress to TA and received feedback.
- **Next Steps:** Continue with webpage and API work, improve design consistency.

Meeting: Sprint Retrospect & Kick off Sprint 3

- **Date:** 21st March 2025
- **Time:** 11:00 AM – 2:00 PM
- **Venue:** Library Room STH503C
- **Key Points:**
    - **Review of Completed Work:** Team reviewed completed tasks, ensuring alignment on project milestones.
    - **CSS Update:** Justin demonstrated CSS updates; merging changes is challenging due to frequent updates in functionality.
    - **Pytest Framework:** Geoff discussed the pytest framework implementation, with plans for further testing and documentation.
- **Next Steps:** Members to continue working on assigned tasks and prepare for the Progress Report due on 23rd March.

Team Contribution

GitHub Contributions



Contributors

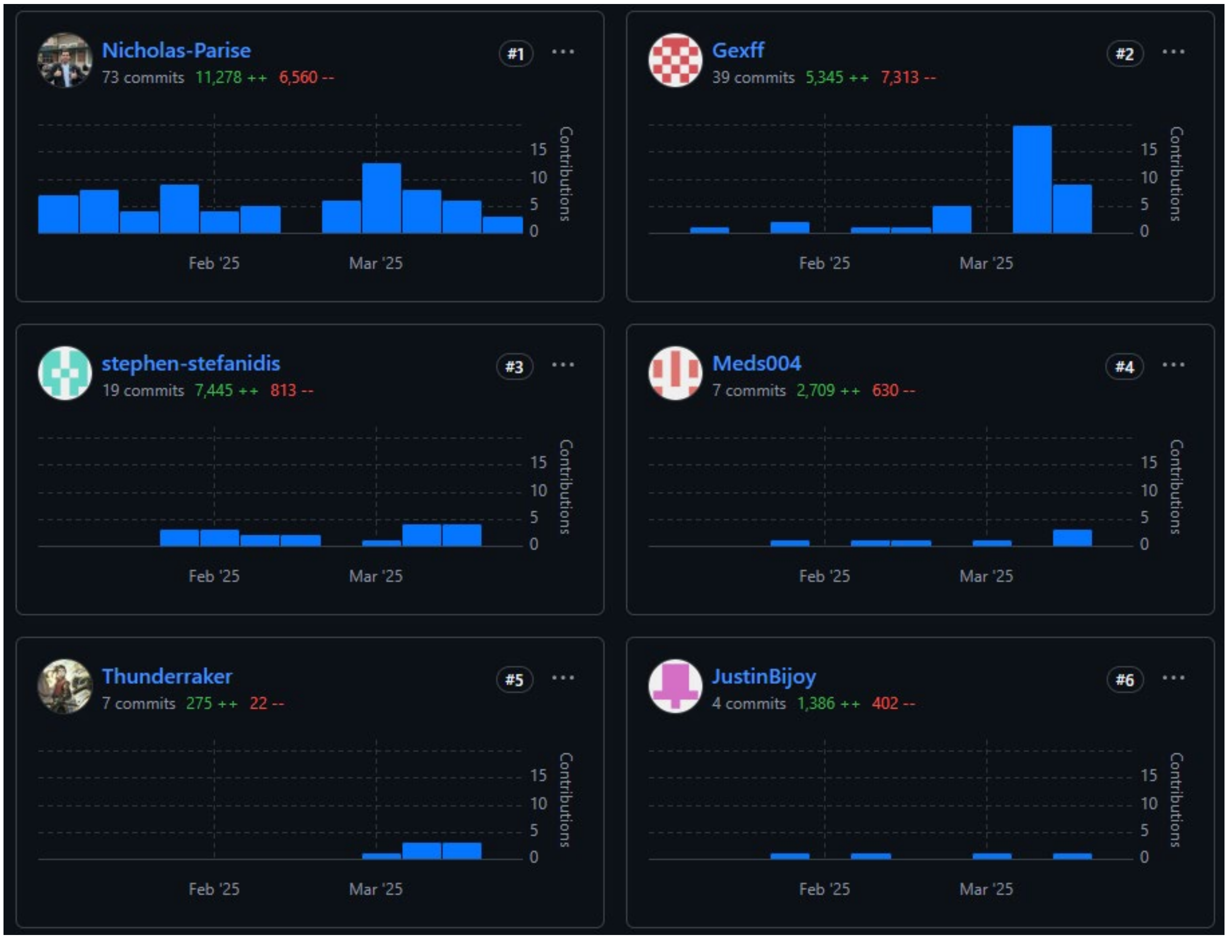
Contributions per week to main, excluding merge commits

Period: All

Contributions: Commits

Commits over time

Weekly from Jan 4, 2025 to Mar 22, 2025



Nicholas-Parise #1
73 commits 11,278 ++ 6,560 --

Gexff #2
39 commits 5,345 ++ 7,313 --

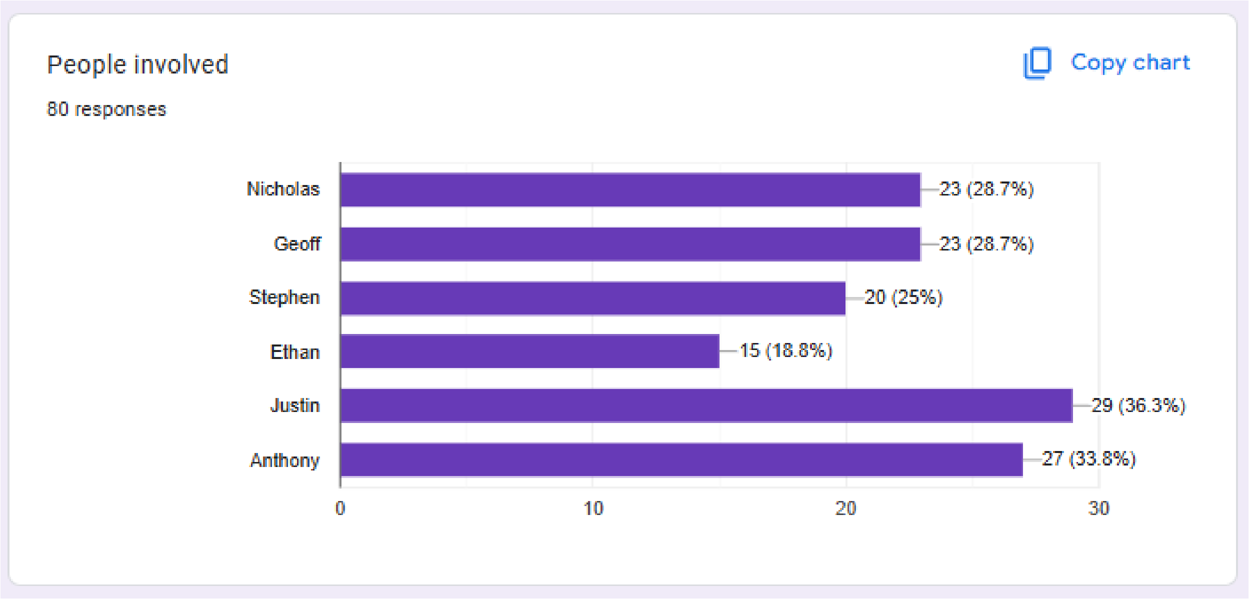
stephen-stefanidis #3
19 commits 7,445 ++ 813 --

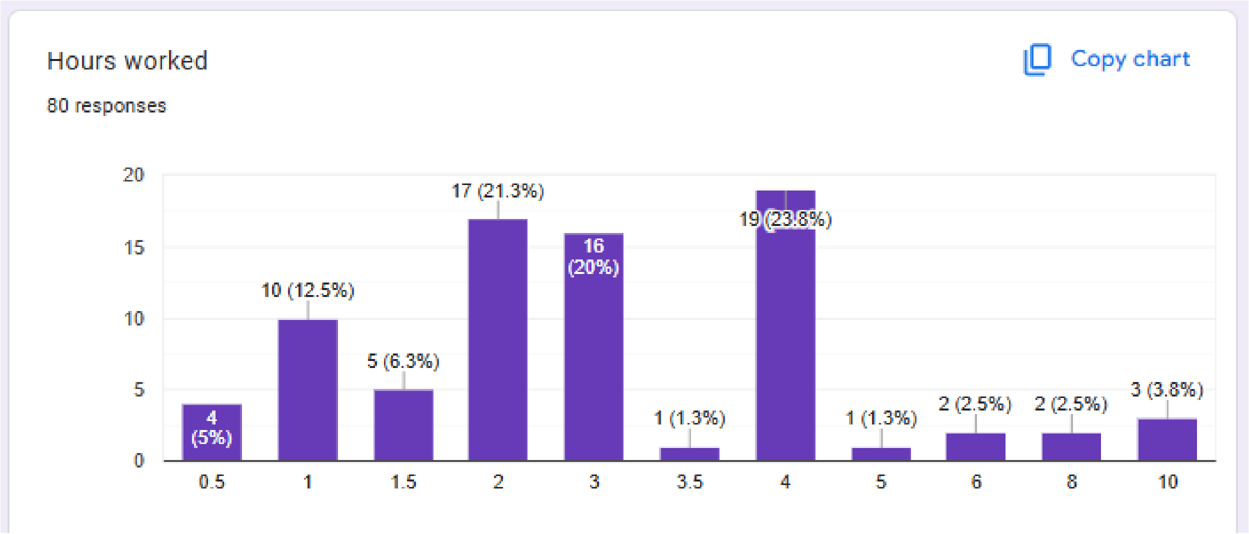
Meds004 #4
7 commits 2,709 ++ 630 --

Thunderraker #5
7 commits 275 ++ 22 --

JustinBijoy #6
4 commits 1,386 ++ 402 --

Work Reporting

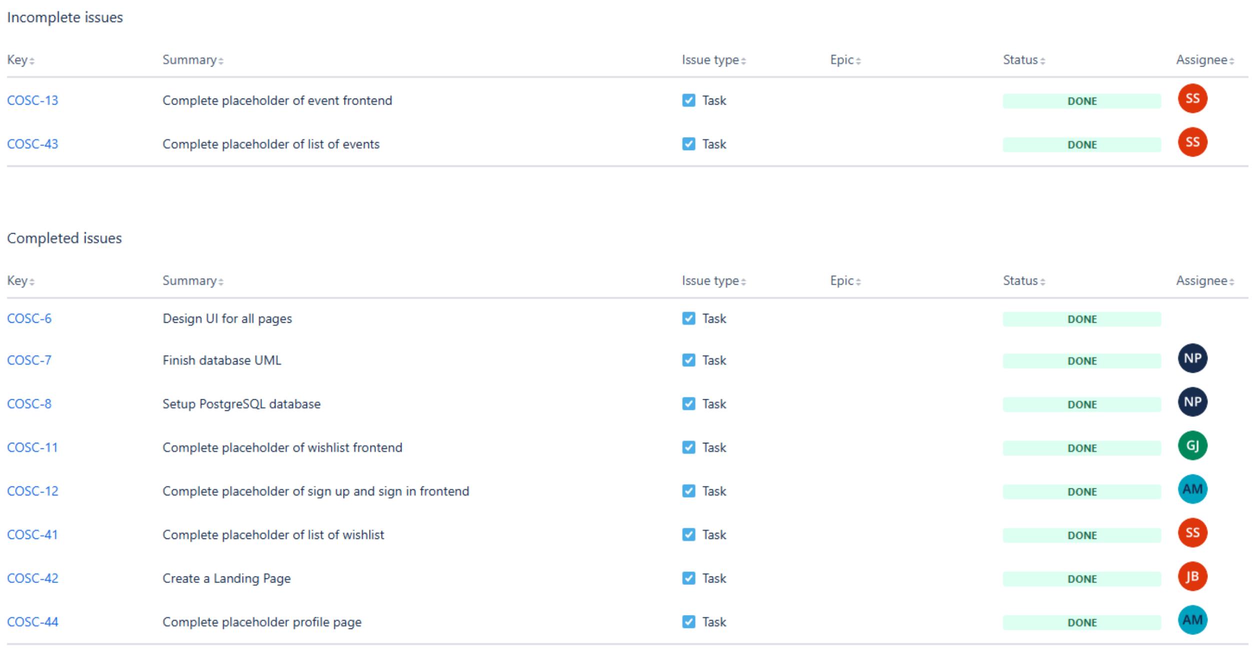




This spreadsheet shows the work each individual has reported, and acts as a nice breakdown of contributions – Contribution Breakdown

Jira Work Assignment

Sprint 1



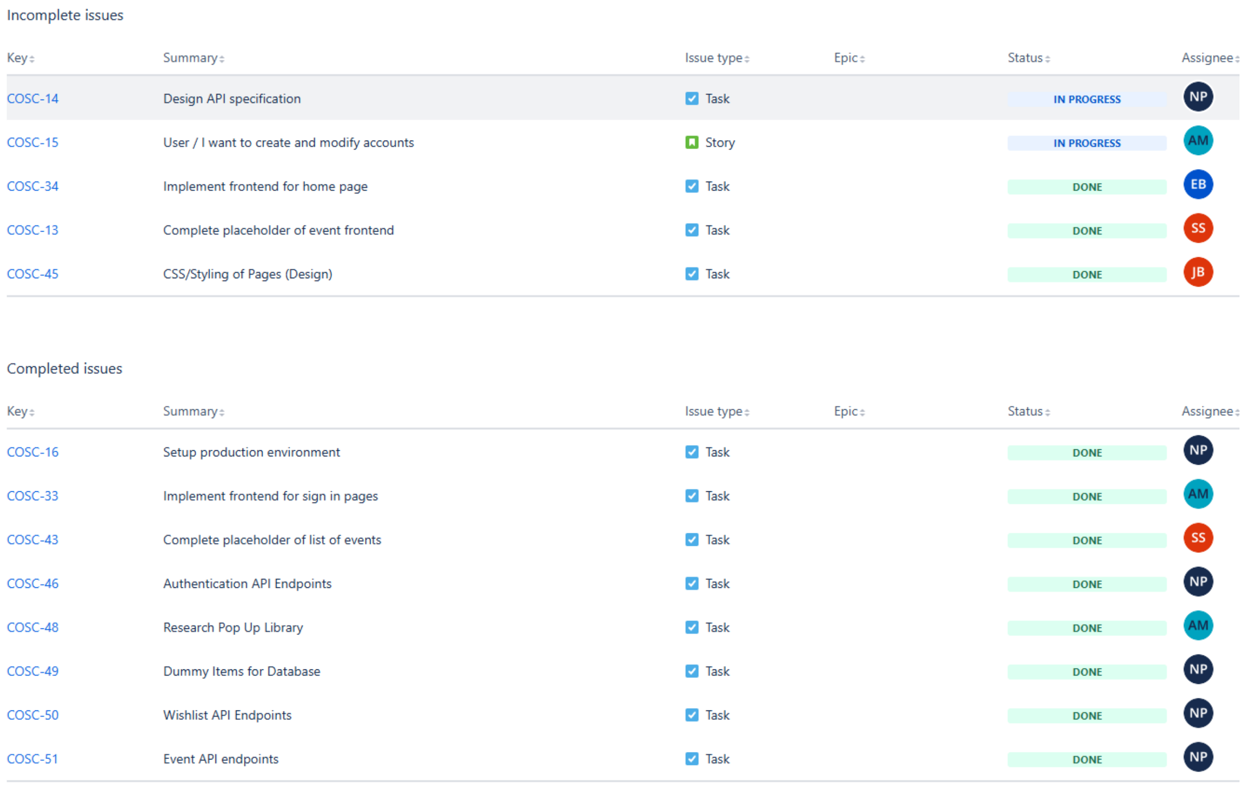
Incomplete issues

Key	Summary	Issue type	Epic	Status	Assignee
COSC-13	Complete placeholder of event frontend	☑ Task		DONE	SS
COSC-43	Complete placeholder of list of events	☑ Task		DONE	SS

Completed issues

Key	Summary	Issue type	Epic	Status	Assignee
COSC-6	Design UI for all pages	☑ Task		DONE	
COSC-7	Finish database UML	☑ Task		DONE	NP
COSC-8	Setup PostgreSQL database	☑ Task		DONE	NP
COSC-11	Complete placeholder of wishlist frontend	☑ Task		DONE	GJ
COSC-12	Complete placeholder of sign up and sign in frontend	☑ Task		DONE	AM
COSC-41	Complete placeholder of list of wishlist	☑ Task		DONE	SS
COSC-42	Create a Landing Page	☑ Task		DONE	JB
COSC-44	Complete placeholder profile page	☑ Task		DONE	AM

Sprint 2



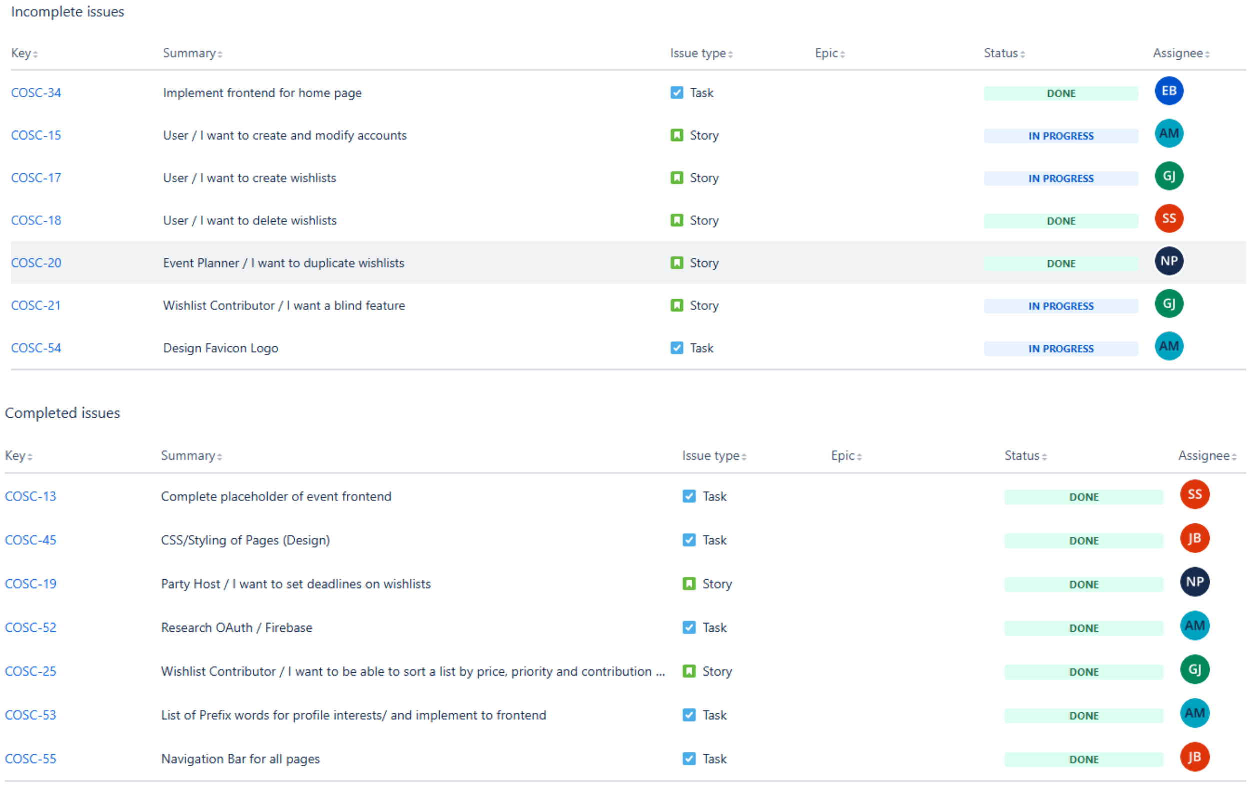
Incomplete issues

Key	Summary	Issue type	Epic	Status	Assignee
COSC-14	Design API specification	☑ Task		IN PROGRESS	NP
COSC-15	User / I want to create and modify accounts	🟩 Story		IN PROGRESS	AM
COSC-34	Implement frontend for home page	☑ Task		DONE	EB
COSC-13	Complete placeholder of event frontend	☑ Task		DONE	SS
COSC-45	CSS/Styling of Pages (Design)	☑ Task		DONE	JB

Completed issues

Key	Summary	Issue type	Epic	Status	Assignee
COSC-16	Setup production environment	☑ Task		DONE	NP
COSC-33	Implement frontend for sign in pages	☑ Task		DONE	AM
COSC-43	Complete placeholder of list of events	☑ Task		DONE	SS
COSC-46	Authentication API Endpoints	☑ Task		DONE	NP
COSC-48	Research Pop Up Library	☑ Task		DONE	AM
COSC-49	Dummy Items for Database	☑ Task		DONE	NP
COSC-50	Wishlist API Endpoints	☑ Task		DONE	NP
COSC-51	Event API endpoints	☑ Task		DONE	NP

Sprint 3



Incomplete issues

Key	Summary	Issue type	Epic	Status	Assignee
COSC-34	Implement frontend for home page	☑ Task		DONE	EB
COSC-15	User / I want to create and modify accounts	Story		IN PROGRESS	AM
COSC-17	User / I want to create wishlists	Story		IN PROGRESS	GJ
COSC-18	User / I want to delete wishlists	Story		DONE	SS
COSC-20	Event Planner / I want to duplicate wishlists	Story		DONE	NP
COSC-21	Wishlist Contributor / I want a blind feature	Story		IN PROGRESS	GJ
COSC-54	Design Favicon Logo	☑ Task		IN PROGRESS	AM

Completed issues

Key	Summary	Issue type	Epic	Status	Assignee
COSC-13	Complete placeholder of event frontend	☑ Task		DONE	SS
COSC-45	CSS/Styling of Pages (Design)	☑ Task		DONE	JB
COSC-19	Party Host / I want to set deadlines on wishlists	Story		DONE	NP
COSC-52	Research OAuth / Firebase	☑ Task		DONE	AM
COSC-25	Wishlist Contributor / I want to be able to sort a list by price, priority and contribution ...	Story		DONE	GJ
COSC-53	List of Prefix words for profile interests/ and implement to frontend	☑ Task		DONE	AM
COSC-55	Navigation Bar for all pages	☑ Task		DONE	JB

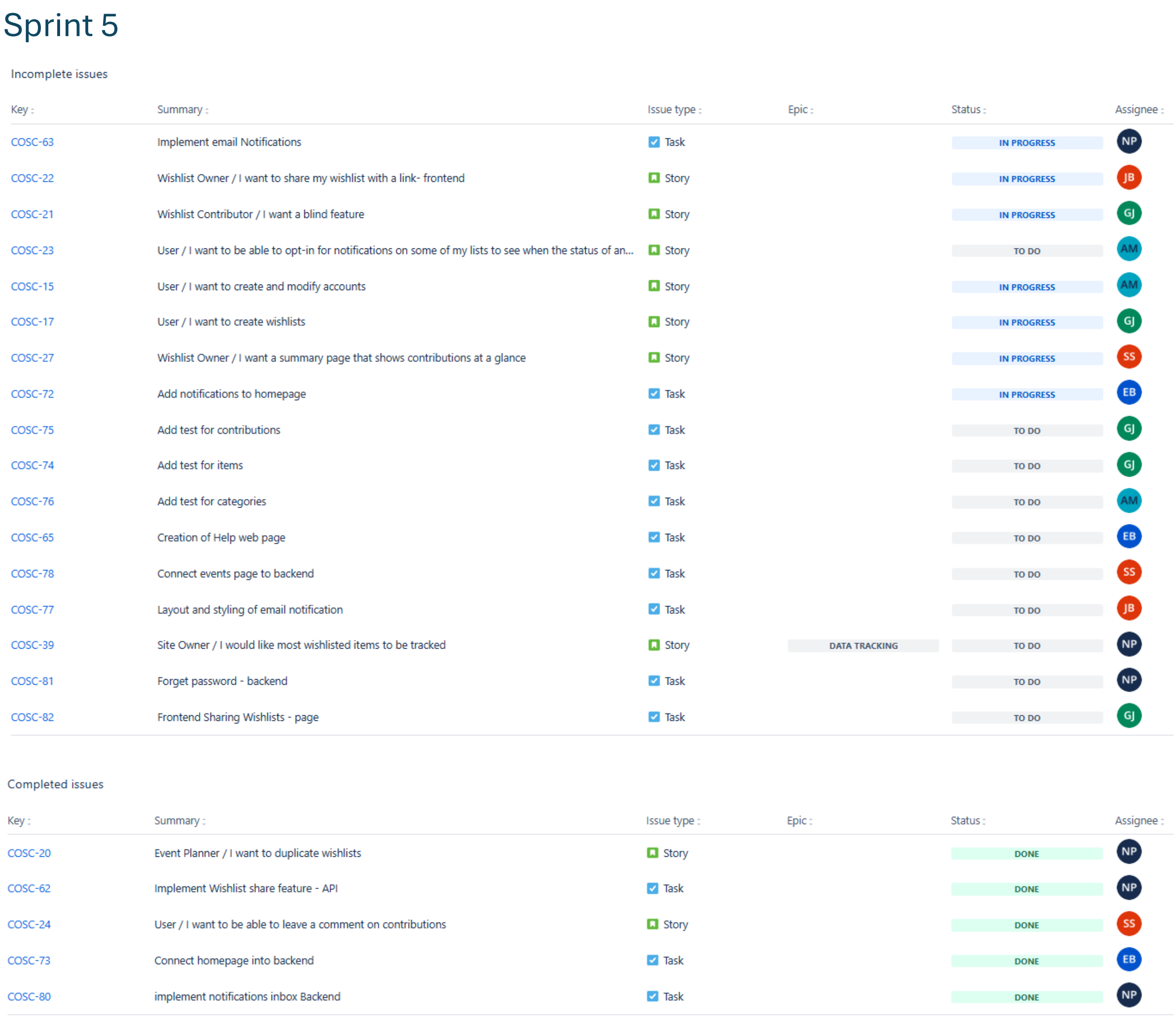
Sprint 4

Incomplete issues

Key	Summary	Issue type	Epic	Status	Assignee
COSC-20	Event Planner / I want to duplicate wishlists	Story		DONE	NP
COSC-21	Wishlist Contributor / I want a blind feature	Story		IN PROGRESS	GJ
COSC-15	User / I want to create and modify accounts	Story		IN PROGRESS	AM
COSC-17	User / I want to create wishlists	Story		IN PROGRESS	GJ
COSC-61	CSS Styling of Pages	Task		IN PROGRESS	JB
COSC-62	Implement Wishlist share feature - API	Task		DONE	NP
COSC-63	Implement email Notifications	Task		IN PROGRESS	NP
COSC-22	Wishlist Owner / I want to share my wishlist with a link- frontend	Story		IN PROGRESS	JB
COSC-23	User / I want to be able to opt-in for notifications on some of my lists to see when the ...	Story		TO DO	AM
COSC-24	User / I want to be able to leave a comment on contributions	Story		DONE	SS
COSC-27	Wishlist Owner / I want a summary page that shows contributions at a glance	Story		IN PROGRESS	SS

Completed issues

Key	Summary	Issue type	Epic	Status	Assignee
COSC-34	Implement frontend for home page	Task		DONE	EB
COSC-18	User / I want to delete wishlists	Story		DONE	SS
COSC-60	API Contributions	Task		DONE	NP
COSC-26	User / I want to set interests on my profile that other people can see	Story		DONE	AM
COSC-64	API Categories	Task		DONE	NP
COSC-36	Implement frontend of event pages	Task		DONE	SS
COSC-70	Write Pytest for end to end testing	Task		DONE	GJ
COSC-71	API add support for arrays	Task		DONE	NP



Sprint 5

Incomplete issues

Key	Summary	Issue type	Epic	Status	Assignee
COSC-63	Implement email Notifications	Task		IN PROGRESS	NP
COSC-22	Wishlist Owner / I want to share my wishlist with a link- frontend	Story		IN PROGRESS	JB
COSC-21	Wishlist Contributor / I want a blind feature	Story		IN PROGRESS	GJ
COSC-23	User / I want to be able to opt-in for notifications on some of my lists to see when the status of an...	Story		TO DO	AM
COSC-15	User / I want to create and modify accounts	Story		IN PROGRESS	AM
COSC-17	User / I want to create wishlists	Story		IN PROGRESS	GJ
COSC-27	Wishlist Owner / I want a summary page that shows contributions at a glance	Story		IN PROGRESS	SS
COSC-72	Add notifications to homepage	Task		IN PROGRESS	EB
COSC-75	Add test for contributions	Task		TO DO	GJ
COSC-74	Add test for items	Task		TO DO	GJ
COSC-76	Add test for categories	Task		TO DO	AM
COSC-65	Creation of Help web page	Task		TO DO	EB
COSC-78	Connect events page to backend	Task		TO DO	SS
COSC-77	Layout and styling of email notification	Task		TO DO	JB
COSC-39	Site Owner / I would like most wishlisted items to be tracked	Story	DATA TRACKING	TO DO	NP
COSC-81	Forget password - backend	Task		TO DO	NP
COSC-82	Frontend Sharing Wishlists - page	Task		TO DO	GJ

Completed issues

Key	Summary	Issue type	Epic	Status	Assignee
COSC-20	Event Planner / I want to duplicate wishlists	Story		DONE	NP
COSC-62	Implement Wishlist share feature - API	Task		DONE	NP
COSC-24	User / I want to be able to leave a comment on contributions	Story		DONE	SS
COSC-73	Connect homepage into backend	Task		DONE	EB
COSC-80	implement notifications inbox Backend	Task		DONE	NP

Appendices

Meeting Minutes - https://drive.google.com/drive/folders/1nsdbSNB-GpEHCatmVDX_s_xBf3spju8h?usp=drive_link

Jira Link - https://nicholasparise.atlassian.net/jira/software/projects/COSC/summary

API Specification Document - https://docs.google.com/document/d/1SM71JXIzTIFZVJ1sR-axeLfNjZRz7TaqCHUoisUTTTc/edit?usp=drive_link

GitHub - https://github.com/Nicholas-Parise/4P02-course-project/

Work Contributions - https://docs.google.com/spreadsheets/d/1eE5I75YznTkIsjcjskV1zBQ9aID18V7N_9SvYZ6pKC4/edit?usp=drive_link

Automated Testing Flow Documentation - https://docs.google.com/document/d/1wd0quA3oHx1mrjeaF-NYpXehc5rbDywUmU-wqXefL1w/edit?usp=drive_link