User Stories	What went well		What needs to be improved and learnings	
	Frontend	Backend	Frontend	Backend
As a grantor or applicant, I want to be able to change the contrast so I can accommodate my vision impairment	Alykhan - Was able to implement monochrome feature on most of the pages relatively quickly.	N/A	Alykhan- Only managed to implement monochrome mode into home page, register, login, forgot password, and reset password, the grantee and grantor side monochrome mode was not implemented	N/A
As a grantor or applicant with protanopia, deuteranopia or tritanopia (types of colour blindness), I want an option to accommodate for my vision impairment	Carter - Using contexts set up previously through the accessibility bar it was simple to make the current mode available to each specific page allowing it to be styled accordingly		Carter - Going through every page was a very very very large amount of work on the grantee side since it was not initially developed to conform to even our dark mode (colour blind modes were not developed yet when the pages were made so this is as expected). I could continuously integrate this on my grantor side, but going back at the end to implement this in every other page took a lot of time. These features should be developed first so they can be continuously integrated everywhere alongside other developlemt and avoid compounding the workload.	
As a grantor or applicant, I want to be able to use screen readers so that I can navigate and use the website effectively	Auric - It was fairly straightforward to use aria-labels and aria-hidden to make tabbing through the page more clear. The stuff that Window's Narrator was saying before the improvements was confusing (e.g. it would say the alt text of images and the label of buttons, so it would say a button's purpose twice), and now it's a lot more easier to understand and follow along. Carter - Many components have good base screen reading capabilities, adding aria-labels and aria-labelledby for a lot of the input fields was less challenging than tedious		Auric - I do not know if Window's Narrator has issues with Firefox, Firefox doesn't like ReactDnD, or ReactDnD isn't up-to-date because ReactDnD and Window's Narrator don't work too well with each other. Whatever draggable item is selected with Narrator enabled, that item can't be dragged with the keyboard even though it works just fine without Narrator. Also, my custom announcements for ReactDnD didn't work (Narrator still said the default annoucements) even though I followed the doc's example. Carter - Select forms were a bit different to make accessible and the shear amount of things that needed to made accessible on the grantee side was a lot of work. If the pages were initially developed with this in mind, the workload could have been decreased	
As a grantor I want to be able to create a new blank grant application form so I can begin making the form		Aahil - Added tests continuously as we implented new endpoints which made it easier to safely add fixes/modifications Ali - Our tests cleanly added and deleted all grant and application objects used accross all the test cases which made it more convenient when running tests while also using the website		Ali - More coordination between frontend and backend teams could have reduced the need to change the data models (and as a result, cut down on the time spent refactoring) Aahil - More consistent documentation for requirements. For example, for filters there was a lot of frontend-backend back and forth to determine and clarify how we wanted the user to be able to filter on grants to apply to. Ben - Some routes had to be scrapped due to them ending up being useless, which is unfortunately because work for them was already been done. Once again just a slight communication issue
(Improve from sprint 2) As an applicant, I want to be able to browse grants pertaining to me so that I can easily find new grants to apply to	Alykhan - worked on grantee dashboard + applied and searched grants integration with the help of backend we were able to get this integrated	Ben - While working with Carter we were able to find many issues with routes on both the grantor and grantee side and fix many bugs before we had to complete the demo	Carter - Being roadblocked by the lack of applications needed to develop the application processing pages on the grantor side, integration of these pages from Alykhan took so long (until March 27) that I did it all myself. He was working on it coincidentally the same time I was but mine was agreed to be used in the final branch Alykhan - another member on the team was also working on the integration at the same time, so we could have saved each other the duplicated efforts. More coordination could have existed between the team.	
(Improve from sprint 2) As an applicant I want to view all the grants I have applied to so that I can see my pending applications	Carter - Integration of the filters wasn't too bad, it was just a matter of a lot of state handling		Carter - Again I had to do all the integration myself even though it wasn't assigned to me since I was being roadblocked and was running out time needed to finish development of the grantor side. A lot of refractoring had to be done since some the pages were not developed under the assumption that a grantee could apply to the same grant multiple times.	
(Improve from sprint 2) As a grantor, I want to be able to close/reopen the application process to new applicants so that I can enforce submission deadlines or have time to review all existing applications	Carter - Implementing the awarded status was easy on the frontend. Some small changes to routes were needed to ensure that the number of winners was being incremented correctly, communicating with the backend team was efficient to get this working	Ben - Aditional routes were added for specific needs which greately improved the ease of integration with frontend	Carter - Some of the route changes could probably have been avoided with a more comprehensive planning session to ensure the routes were originally made with the complete knowledge of what the frontend needed.	
(Improve from sprint 2) As a grantor, I want to be able to choose an applicant(s) to receive a grant so I can move to the next stage in providing the grant.	Carter - A view answers component made by Auric was useful in filling out this page and facilitating the review process. Adding buttons to change the application status was easy with consistent communication with the backend team.	Ben - Same as previous, dditional routes were added for specific needs which greately improved the ease of integration with frontend	Carter - Revoking accepted applications and accepting rejected applications all according to the current grant status if its awarded or not was a bit more complex than expected and caused some unintended behaviour on tracking winners, but ultimately is was quickly fixed by the backend team	