Approach:
To me exploratory testing means finding out how the system behaves under unexpected behaviours (expected behaviours should be checked through automation). My approach is:

- Step through as a User. Concentrate on functionality and visuals but with a mindset of different types of Users (e.g. a User on mobile, a User trying to break the system).
- Review any console errors or warnings and view the underlying code. With context from the User flows I like to see how things are working and components have been constructed.

Scenarios I tried:

- Allow view and edit on my page (from everyone)
- Allow edits on my page (by specific users)
    - Add same user from search twice with different permissions
    - Remove users from user search box with small x
    - If a user has been added from search, can I search their name again, remove them from the bottom list, then add them again over the top with different permissions
    - Add as many users from user search as possible
- Allow viewing of my page (by specific users)
    - If I add myself with View permission will that overwrite my existing Edit permission
- Hide my page from everyone except me
- Inspect permissions of a user
- Edit permissions from Mobile
    - Mobile web, no icon cannot access modal
    - Mobile web desktop mode
    - Mobile app

Bugs I found:

- For the ‘Add as many users from user search as possible’ scenario, I added 60 users then the dropdown disappeared and I could not add any more. However my directory is much larger than 60. I can add a user not in the initial list so connectivity to the directory is working. So the user search dropdown doesn't generate more users when the bottom is reached OR the backend is not providing more results to add on. Although manually adding more than 60 users is an unlikely use-case, I would check this with the team.
- Mobile web desktop mode: Functionality works but with some odd behaviours. Tooltip shows by default, ‘Can view/edit button’ text is cut off. However since I essentially forced the desktop version against Confluence's will it's probably not supported but again would check.
- Mobile App flow is a bit different, opening a page I have to edit first then a lock icon appears. Restriction options are different: For ‘Only some can view and edit’ there doesn't seem to be any user selection, just shows me in the list. This might be a design decision to only allow viewing through app but editing permissions must be through desktop. Again, would clarify with the team.

Further testing areas:

- Before shipping this to millions of users, I would like to ensure:
    - Changing permissions on one side may show success in the UI but has it actually changed the other users’ accounts ability to view or edit (as I have no visibility of the backend).
    - Performance is acceptable under load. If one user changes the permissions of a large corporate group it could be thousands of requests in the backend.
    - Browser network calls are secure and not exposing any sensitive user information
    - Visuals are not affected by non-English localisations (if we are in those markets)
    - Visuals look correct and functionality works across supported browsers and devices
    - Accessibility requirements are met