

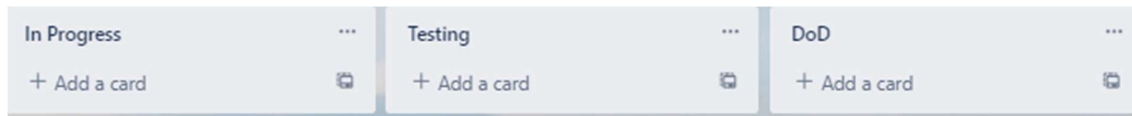
Testing

The testing procedure we followed varied depending on the state of the project and whether the development team were in a sprint or not.

During sprints:

During sprints, we followed a standard Agile approach using our Trello board. User stories and tasks would be assigned to developers, and while work proceeded, would be moved to the 'In Development' column. Once the developer(s) assigned finished, they would be moved to the 'Testing' column, where another developer that had not worked on that feature would pull their work and test it. Meanwhile, the developer(s) working on that feature would open a PR on GitHub to merge into main.

If the tester(s) are happy with the work, they'd complete the merge and leave a comment outlining as such. The tester would also then move the user story or task on the Trello board into the 'DoD' column. If there was any further confusion, the testers would reach out directly to the developers who worked on the user story or task and get clarification, usually in a discord call. Otherwise, if some issues are clearly still there, the PR (pull request) is left with comments detailing as such, and the developer informed of things that must be fixed first (or reached out to directly instead).



Example of a PR sent back to the original developer below:

Sigma 015/upload data read app #2

In Progress Metarock merged 16 commits into [main](#) from [sigma-015/upload-data-read-app](#) on Apr 14

Conversation 2 · Commits 16 · Checks 8 · Files changed 18

Metarock commented on Mar 27

Issue:
Currently we are able to make a HTTP POST request to the database when uploading a share. We need to allow update request to be used.

Solution:
Implement HTTP PUT request in Axios method and ensure that the application when to request a PUT or POST method.

Risks:
Need to thoroughly tests this as we are unsure the extent of our error catching.

Reviewed By:
Peter, Chris, Jose

Metarock added 9 commits 3 months ago

- added GET method for apiShares
- setup fund upload data table
- setup for update shares\$00; setup for upload funds
- distinguish which share is in db
- add the ability to upload and update shares
- display pop up notification if upload is success or not
- editor changes
- fixed error throw

added cypress for testing in the near future

8c4253c

Metarock requested review from **PScandle**, **jose-vs** and **castehin** 3 months ago

Metarock assigned **Metarock** and **jose-vs** on Mar 27

jose-vs approved these changes on Mar 28

[View changes](#)

jose-vs left a comment

looks great, updating shares works great, though there are codes for uploading funds that are still not functional

jose-vs requested changes on Mar 28

[View changes](#)

jose-vs left a comment

There seems to be an error when reading in empty cells in excel files. Also, the method for updating or uploading might be inefficient yeah yeah you already know easy brother

Example of a successful PR with a comment denoting as such:

Sigma 002/configure get and post requests for sd gs #1

Metarock merged 4 commits into `main` from `sigma-002/Configure-GET-And-POST-Requests-For-SDGs` on Sep 27, 2021

Conversation 1 · Commits 4 · Checks 0 · Files changed 14

castehlin commented on Sep 27, 2021

Issue:
Having the 17 SDGs in the backend so that they may be filled with user quiz data and viewed later.

Solution:
17 new columns were added to the user table. When a user is made, these 17 columns are created in the same row for that user. They are null for now, but are ready to be updated via a PUT request.

Risk:
The method chosen was not the preferred one. It would be better from a database design perspective to have the SDGs in a separate table and linked to the user table via the `user_id` as a foreign key. I wasn't able to figure out how to do that yet though - having the SDG table recognize a valid user request was something I am not able to do yet. The plan is to revisit this once I know how to.

Reviewed By:
[Edit the message as to who reviewed it here]

castehlin added 4 commits 9 months ago

- get and post methods set up for a test table. Need to switch over to ... `ba1b116`
- before adding field to user table.. `d8b9c5e`
- sdg fields added to user table successfully. GET and PUT only, fields.. `c6dbadb`
- Auth/user table now has sdg fields. GET and PUT requests functional. `2c624af`

castehlin requested a review from Metarock 9 months ago

Metarock approved these changes on Sep 27, 2021 [View changes](#)

Metarock left a comment

LGTM!

Metarock merged commit `efae89` into `main` on Sep 27, 2021 [Revert](#)

After sprints (before Heroku implementation):

After sprints, we would show our deliverables to our client and mentor in meetings locally on one of the development team's laptops. As the nature of our deliverables for each sprint generally revolved around showing data output based on some data input (whether it be ordering Sustainable Development Goals (SDG) in a quiz, interpreting dummy data and comparing it to the output SP data, or using real or "real enough" data depending on the stage of the project) our client served as a further testing step once it became time to show her the deliverables at the end of the sprint. Her feedback could result in moving onto new features, sending features back for further development, or changing the scope for future sprints depending on what was revealed at the end of these sprints.

After sprints (Heroku deployment implemented):

After all features in a sprint had been developed, the development team would deploy their work to the Heroku test deployment for our client to test our deliverables on. This became more relevant once the data upload app had been implemented, as our client then had the ability to test her own data entered in the share and fund excel templates, we designed for her. This would enable the

client to give feedback on things we had not thought about during our earlier presentations that only used the data we entered ourselves, rather than data of her own.

Editable excel templates given to client to allow for whatever raw data she entered to be used in the data upload app, which could then be viewed on the finappster frontend recommendations page to see if it showed as expected.

	Share/company name	ISIN code	No poverty	Zero hunger	Good health & wellbeing	Quality education	Gender equality	Clean water & sanitation	Affordable & clean energy	Decent work & economic growth	Industry innovation & infrastructure	Reduced inequalities
1	BHP	AU000000BHP4	35.33333%	44.44444%	11.11111%	33.33333%	1.11111%	22.22222%	22.22222%	1.11111%	21.21212%	21.21212%
2	Commonwealth Bank of Australia	AU000000CBA7	35.33333%	44.44444%	11.11111%	33.33333%	1.11111%	22.22222%	22.22222%	1.11111%	21.21212%	21.21212%
3	CIL	AU000000CIL3	42.42424%	47.47474%	12.12121%	34.34343%	1.11111%	23.23232%	23.23232%	1.11111%	22.22222%	22.22222%
4	Intel Corporation	AU000000INT4	41.21212%	46.21212%	11.11111%	33.33333%	1.11111%	22.22222%	22.22222%	1.11111%	21.21212%	21.21212%
5	Teclia Corporation	AU000000TEC5	41.21212%	46.21212%	11.11111%	33.33333%	1.11111%	22.22222%	22.22222%	1.11111%	21.21212%	21.21212%
6	Charles Schwab Corporation	US000000CSH6	38.38383%	43.38383%	10.38383%	32.38383%	1.11111%	21.38383%	21.38383%	1.11111%	20.38383%	20.38383%
7	EOO Resources, Inc.	US000000EOO7	38.38383%	43.38383%	10.38383%	32.38383%	1.11111%	21.38383%	21.38383%	1.11111%	20.38383%	20.38383%
8	Reiter Building	US000000REI8	38.38383%	43.38383%	10.38383%	32.38383%	1.11111%	21.38383%	21.38383%	1.11111%	20.38383%	20.38383%
9	Goldman Sachs	US000000GS10	35.35353%	40.35353%	10.35353%	30.35353%	1.11111%	20.35353%	20.35353%	1.11111%	19.35353%	19.35353%
10	Medison Energy	US000000MED11	35.35353%	40.35353%	10.35353%	30.35353%	1.11111%	20.35353%	20.35353%	1.11111%	19.35353%	19.35353%

	Fund Name	Fund Identifier	Percent Of fund	ISIN Code
1	ASX	FN012194	12.121212	AU000000BHP4
2	ASX	FN012194	29.512121	AU000000CBA7
3	ASX	FN012194	39.421142	AU000000CIL3
4	ASX	FN012194	4.644671	AU000000INT4
5	ASX	FN012194	6.745217	US000000CSH6
6	ASX	FN012194	2.212121	US000000EOO7
7	FT-ASX	FN009123	18.114214	NZ000000REI8
8	FT-ASX	FN009123	24.809102	US000000GS10
9	FT-ASX	FN009123	3.991211	AU000000MED11
10	FT-ASX	FN009123	21.435564	AU000000BHP4
11	FT-ASX	FN009123	39.311532	AU000000CBA7
12	FT-ASX	FN009123	12.411112	AU000000CIL3
13	VSI-ASX	FN009129	23.583921	AU000000BHP4
14	VSI-ASX	FN009129	3.789622	AU000000CBA7
15	VSI-ASX	FN009129	11.571112	AU000000CIL3
16	VSI-ASX	FN009129	1.811331	US000000CSH6
17	VSI-ASX	FN009129	5.115721	US000000EOO7
18	VSI-ASX	FN009129	1.411424	US000000GS10
19	VSI-ASX	FN009129	18.214132	AU000000BHP4
20	VSI-ASX	FN009129	6.802422	AU000000CBA7
21	VSI-ASX	FN009129	11.325752	AU000000CIL3
22	VSI-ASX	FN009129	3.212121	AU000000INT4
23	VSI-ASX	FN009129	42.100011	AU000000CSH6
24	VSI-ASX	FN009129	16.121422	NZ000000REI8

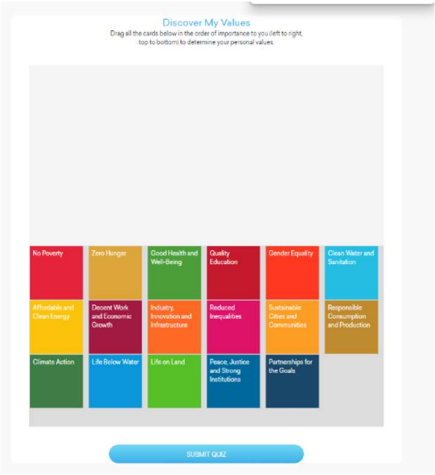
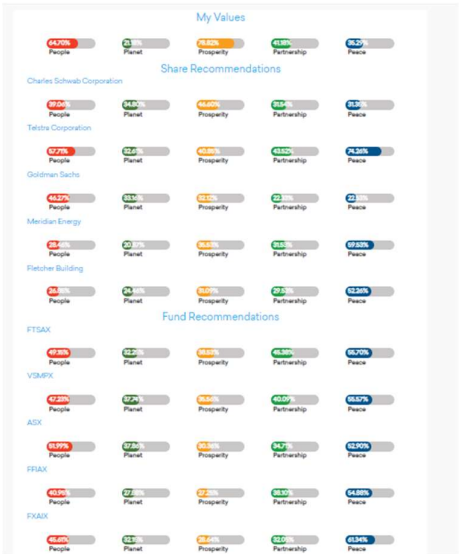
Data upload app for client to upload test data to.

SELECT A SHARE FILE

SELECT A FUND FILE

Delete all data

Quiz for client to test different combinations of SDGs to analyze 5P output.



Quiz and uploaded share and fund data output. Client could then compare this to her expected output that she could calculate herself to determine if it is how she wanted it to be.