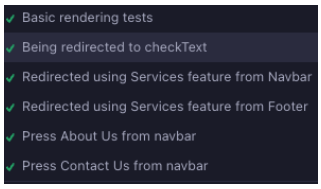
Sprint retrospective, Milestone #5

User story	Task	Assignee/s	Estimated effort	Actual effort	Done	Notes
Integration tests for checkURL	#70	Vlad Nitu	2h30m	2h30m	Completed	Set-up `cypress` as a integration testing tool & Test Runner; configure codestyle to work with `cypress`; Integration test `checkURL` by performing 6 different actions from `checkURL` page:  Augment test suite after @rtoader added his tests, as he had some interesting ways to assert diff. behavior. Created a custom assert method to check for `isInViewport` of different rendered elements
Integration tests for checkText	#72	Rares Toader	3h	3h	Completed	Worked with cypress to add tests for the `checkText` page to include most flows. Had some difficulty with finding some HTML elements, but then added some attributes for better observability(e.g. `data-testid`).

Integration tests for checkTwoTexts	#78	Matei Mirica	5h	4h	Done	Wrote integration tests for the ‘checkTwoTexts’ page using ‘cypress’.
Persist URLs in the background	#52	Matei Mirica	6h	4h	In-progres s (moved to Sprint 6)	When checking an url for similar articles, it will be stored in the database (if it is not already there). The article was persisted before the similar articles were retrieved. This issue aims to store the article “in the background”, after the similar articles are returned. I am doing this using ‘Celery’. However, I encountered some issues with the setup, but now it works. TODO: add running the RabbitMQ server and ‘Celery’ worker from the pipeline when running tests.
Populate the database until 10-100k and test whether the algorithm is feasible on such amount of data.	#68	Rares Toader & Vlad Nitu	10h	5h (3h Vlad Nitu + 2h Rares Toader)	In-Progres s (started in Sprint 5, will be continued in Sprint 6)	Rares & Vlad: Helped with persisting the data in MongoDB, upgrading the clusters and then some troubleshooting the storage space used by MongoDB, searching for alternatives. Constantly profiled how efficient our queries are on the database at around: 1k, 2k, 5k and 10k documents.

						Removed `date` attribute as it was redundant (and thus observed a boost in memory-storage performance) Tested the algorithm by trial-and-error to see if it is feasible or not
Deploying the web application	#58	Rares Toader	14h	1h30m	Postponed	Did research on how to deploy on Heroku, tried some experiments on a separate branch. Researched what is better: deploying as a single application, or deploying the `backend` module separately than the `frontend` module. Had some difficulty for syncing the config files with heroku. This issue is postponed for the start of last sprint.
As a user, I desire a system that checks the originality of an article based on its publication date, with the aim of obtaining ownership of the article in question.	#3	Rares Toader & Vlad Nitu	10h	10h (5h Rares + 5h Vlad)	Completed	Mainly @vvladpetru on frontend and @rtoader on the backend, but we've discussed all the design decisions and paired programmed throughout the process. -> Check !65 for a thorough explanation of the feature. Rares: -> Implemented a reponse entity and a encoder for it, for the 'checkTwoUrls' response.

						-> Edited the compareURLs endpoint to better suit the feature trying to be implemented, by adding dates and a ownership variable. -> added and changed existing tests, in order to preserve the high branch coverage in the ‘backend’ module. Vlad: -> Implemented `Ownership` React component and integrated with the main `checkTwoURLs` component; took both publish dates from backend server and displayed them in our UI. -> When we are able to compute the ownership of the two news articles, we also display an arrow to show which article owns the article (probably) -> the left or the right on -> fully unit tested Overview component, in order to preserve the high branch coverage in the frontend module.

`Persist` phase of a news document running slow	#74	Vlad Nitu	-	1h15m	Completed	Instead of using Python's `list comprehension` feature which seems pretty slow, go for `distinct()` which returns the same result (an array/list of hash values) *!* - the `set(...)` operation was slow, but is needed as a trade-off for fast intersection (`&`) and union (`|`) operations on sets. However, this solution seemed to decrease the time it takes to persist a news document (in the current state of the database: around 2000 documents in the database and ~40k hashes value) from 2m30s to 5s :D
Add profiler for Django backend to measure performance of algorithm	#76	Vlad Nitu	1h	4h15m	Completed	While still encountering performance issues with our algorithm and we want to be as efficient as possible in all parts of the backend code, we need a way to profile / measure how fast each line of code / HTTP request is. Thus, we need to measure how many (m)s different batches of operations take

						Integrated `django-silk` 3rd party lib to start a new server on a different port of the local host: that profiles not only the HTTP request calls but also the backend Python method used in these HTTP requests (i.e: helper methods) See !59 for a more detailed description
Chain of responsibility pattern implementation	#65	Diana Micloiu	5h	5h	Completed	Started by creating some UML diagrams for the Chain of Responsibility pattern that mimic the behavior to be implemented. Refactored the persist functionality to be completed by using the chain created.

Integration tests for compareURLs service	#79	Diana Micloiu	5h	4h	Completed	Wrote integration tests for the compareURLs page using ‘cypress’.
Loading Circle for all Services	#59	Cristian Ciacu	2h	2h	Completed	Worked without any problems

Help Me Page	#69	Cristian Ciacu + Rares Toader	12h	20h	Completed	Help page itself did not imply any serious problems, except for the +, - animation which was not working as expected. Then, the unit tests were not working, but in the end Rares helped out, increasing the test coverage to 100%.
E2E testing for Help page	#77	Cristian Ciacu	1h	1h	Completed	Nothing special.