

News Finder

Gekozen technologieën

Back-end

Flask

<https://flask.palletsprojects.com/en/2.2.x/>

Het Flask framework in Python werd aangeraden door de professor. Het flask framework heeft heel veel voordelen, waaronder een goede integratie met databanken, en het is een zeer schaalbaar framework. Het kan een groot aantal verzoeken verwerken als het bedrijf met de tijd opschaaft. Een goed voorbeeld hiervan is Pinterest, dat overgestapt is van Django naar Flask en nu miljarden verzoeken per dag verwerkt.

Prisma

<https://prisma-client-py.readthedocs.io/en/stable/>

Prisma Client Python is gebaseerd op de populaire Prisma ORM. Naast een duidelijke en uitgebreide documentatie, biedt het ook de mogelijkheid om migrations te gebruiken. Het grootste nadeel is momenteel dat de Prisma client voor Python slechts door één persoon gemaakt is en onderhouden wordt.

Pyright

Pyright is een static-type checker voor Python. Het dient als extra hulp tijdens het schrijven van code en laat ons toe code te checken als integration job.

Pytest

<https://docs.pytest.org/en/7.2.x/>

Pytest is een test framework om gemakkelijk een python project te testen. Het laat toe om testen op een eenvoudige manier te schrijven en heeft veel support voor flask applicaties.

Front-end

React

<https://reactjs.org/>

Door React te gebruiken wordt het makkelijk om de front-end reactief te maken en op te delen in kleinere stukken. Het maakt het ons gemakkelijk om verandering in de databank

snel te kunnen weergeven op de front-end zonder nieuwe HTML te moeten genereren, alles is namelijk opgedeeld in components.

Vite

<https://vitejs.dev/>

Vite gebruiken we als build tool voor het front-end deel van onze applicatie. Vite maakt het lokaal ontwikkelen evenals het bouwen van de frontend zeer eenvoudig. Vite heeft namelijk ingebouwde ondersteuning voor typescript. Hierdoor moeten we deze code niet eerst omzetten naar javascript. Ook heeft vite een automatisch refresh mechanisme, hierdoor worden veranderingen in de code meteen weergegeven. Dit ook met eventuele errors, vite heeft namelijk een robuust plugin systeem dat integratie toelaat met type checkers en linters.

Typescript

<https://www.typescriptlang.org/>

Door Typescript in plaats van de gewoonlijke Javascript te gebruiken garanderen we type-safety.

React-Bootstrap

<https://react-bootstrap.github.io/>

Door React-Bootstrap wordt het makkelijk om de stijl van onze website aan te passen met minimale kennis van css. Door zijn populariteit is bootstrap goed gedocumenteerd en zijn antwoorden op vragen en problemen makkelijk te vinden op het internet.

RSS scraper

Voor de RSS scraper hebben we enkele vereisten. Deze moet type-safe zijn en gemakkelijk uit te breiden op basis van de syntax van nieuwe RSS feeds. Ook zoeken we een implementatie die op een efficiënte manier kan omgaan met een enkele core. Dit aangezien we docker gebruiken als execution environment (zie docker voor verdere uitleg).

Rust

<https://www.rust-lang.org/>

Bijna alle vereisten worden ingevuld wanneer we rust gebruiken en het bijbehorende ecosysteem. Rust zal ons namelijk garanderen dat wanneer we alle, door de rust compiler gegeven, errors behandelen, onze applicatie zeer stabiel zal zijn. Daarbovenop laat het rust ecosysteem toe om nieuwe specificaties snel en gemakkelijk te kunnen implementeren bovenop de bestaande code. Tot slot is rust een compiled language gebaseerd op de llvm toolset. Dit maakt dat onze applicatie niet alleen stabiel is, maar ook nog eens alle performance mogelijk uit die enkele core zal halen.

Prisma

De RSS scraper heeft toegang nodig tot onze database. Aangezien we prisma gebruiken kunnen we net als bij de server een nieuw target toevoegen dat code genereert om met onze databank op een volledige type-safe manier te kunnen omgaan.

Cron

<https://crontab.guru/>

Aangezien de RSS scraper een alleenstaande applicatie is, is het nodig dat deze continu blijft draaien. Dankzij cron kunnen we onze applicatie inplannen in een schema en deze bijvoorbeeld elke 10 minuten alle feeds laten opvragen.

Similarity Checker

Het doel van de similarity checker is om artikels van verschillende nieuwsbronnen te groeperen wanneer ze over hetzelfde onderwerp gaan. Als gebruiker wil men namelijk niet zes keer hetzelfde artikel voorbij zien komen. Ook is het nu voor de gebruiker gemakkelijk om informatie te vergaren van verschillende bronnen.

Tf-idf en cosine similarity

Voor onze mvp hebben we ervoor gekozen het groepering algoritme simpel en naïef te houden. Dit laat ons toe later uitgebreider te testen en om deze sprint het tempo hoog te houden. We gebruiken het tf-idf algoritme om artikels om te zetten naar vectoren. Hierdoor kunnen we puur mathematische testen toepassen op wat onderliggend enkel tekst is. We vergelijken elk artikel tegen elk ander artikel om deze te groeperen.

Continuous integration and deployment

CircleCI

<https://circleci.com/>

CircleCI garandeert de volgende gebruiker van de code dat wat eerder is geïmplementeerd ten minste werkt op een linux machine. Aangezien onze groep veelal in linux werkt garandeert dit ons de correctheid en kwaliteit van de al geschreven code. Ook laat dit ons toe om in kleinere fragmenten code af te werken. De integratie van deze code wordt namelijk automatisch getest en gevalideerd.

Docker (Compose)

<https://www.docker.com/>

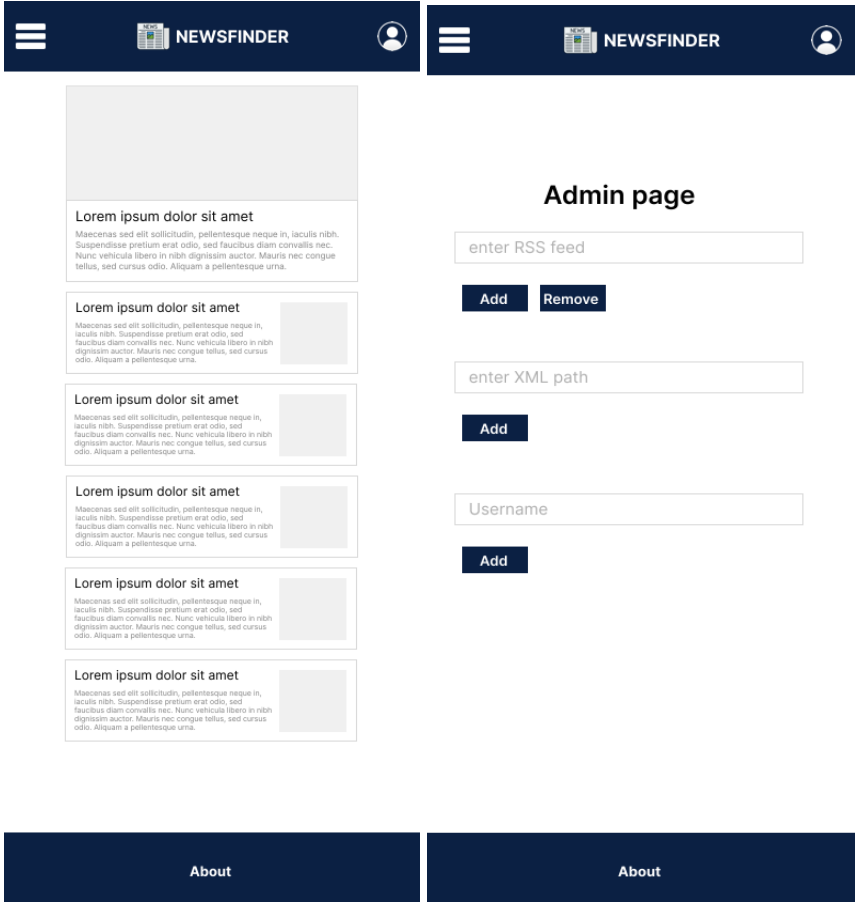
Bovenop CircleCI gebruiken we docker en docker compose om onze applicatie tot leven te wekken. Docker laat ons toe om elk deel van de applicatie te isoleren en gemakkelijk op te schalen indien nodig. Een extreem handig gevolg van deze docker-compose file is dat de developer workflow zeer simplistisch wordt. Men moet maar één commando uitvoeren op een machine met docker geïnstalleerd en onze volledige applicatie kan lokaal gebruikt worden.

Definition of done

- Is gedocumenteerd.
- Getest door de schrijver(s).
- Getest door CircleCI op integratie en unit tests.
- Is nagelezen/getest en goedgekeurd door minstens twee andere personen.

Mockups

Mobile



Admin page

[Add](#)[Remove](#)[Add](#)[Add](#)

Register page

[Sign up](#)

already have an account? [Log in](#)

[About](#)[About](#)

Login

Login

Create account

Desktop

Admin page

Add

Remove

Add

Add

About

Register page

Sign up

Already have an account? [Log in](#)

About

Category

Category

Category

Category

Category

Category

Lorem ipsum dolor sit amet.

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet.

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

About

Login

Login

Create account

About Us

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas sed elit sollicitudin, pellentesque neque in, iaculis nibh. Suspendisse pretium erat odio, sed faucibus diam convallis nec. Nunc vehicula libero in nibh dignissim auctor. Mauris nec congue tellus, sed cursus odio. Aliquam a pellentesque urna. Phasellus dictum leo eget vulputate scelerisque. Vivamus viverra et nisi et pulvinar. Proin non massa non tellus consequat elementum et sollicitudin lorem. Quisque iaculis at tellus et volutpat. Praesent sed nunc ut elit pretium auctor id vel urna.

Sed locus lorem, porttitor ut neque nec, vehicula semper felis. Morbi ut aliquam massa. Id mollis lectus, integer ullamcorper nunc vel ligula fegat molestie id vel ex. Etiam congue, nisi ut amet ultrices tristique, sapien est hendrerit libero, et facilisis neque dui sit amet massa. Sed ex lacus, interdum ut sodales eget, volutpat lacus enim. Nulla imperdiet consectetur libero, eu lorem ante. Maecenas sed hendrerit justo. Vivamus tristique lectus vitae aliquam venenatis. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Quisque vel augue at sem fegat pharetra. Praesent elementum, nisi vel maximus posuere, justo lorem rhoncus ante, et commodo nunc dui id quam. Donec tristique malesuada tortor egestas fringilla. Nulla euifend est a vestibulum accumsan. Maecenas convallis odio volutpat ligula vestibulum, elementum dapibus ante varius.