Dit onderzoek maakt gebruik van

* <https://tiptap.dev/docs/editor/getting-started/install/react>
* <https://github.com/mixmark-io/turndown>
* <https://github.com/cure53/DOMPurify>
* <https://github.com/markedjs/marked>

# Conclusie

Het gebruik van een rich tekst editor is een goed idee en is vrij gemakkelijk te installeren.

Voor gebruik van de rich tekst editor zijn de volgende dependencies nodig:

* Tiptap: een wrapper van ProseMirror. Tiptap zorgt ervoor dat de tekst aangepast wordt op basis van de controls, en geeft daarna HTML terug voor die tekst.
* Turndown: Zet de HTML om in markdown. De teksten worden in markdown opgeslagen in de database, dit scheelt namelijk veel karakters.
* Marked: Zet de markdown om in HTML. Dit zorgt ervoor dat het laten zien kan worden in de browser, na het ophalen vanuit de database.
* DOMPurify: Zorgt ervoor dat de HTML gesatineerd wordt, om XSS-attacks te voorkomen.

### Invoerveld

Van Tiptap gebruiken we de ‘starter-kit’ en ‘underline’ extensions. Dit zorgt ervoor dat we de onderstaande opties kunnen gebruiken:

Afbeelding met tekst, schermopname, lijn, Lettertype

Automatisch gegenereerde beschrijving

Figuur Tekst editor

Voor het testen heb ik een component aangemaakt (components/RichTextEditor.jsx). Deze ziet eruit zoals in figuur 1 te zien is.

Dit component heeft een onSave functie als prop, die ervoor zorgt dat de tekst wordt omgezet naar markdown door middel van Turndown en naar de hogere component wordt gestuurd. De hogere component is er verantwoordelijk voor om de data op te slaan.

### Tonen vanuit de database

Om markdown terug te zetten naar HTML, maken we gebruik van Marked. Nadat het is omgezet naar HTML, gebruiken we DOMPurify om de tekst te sanitizen. Zie hieronder een voorbeeld van hoe die transformatie eruitziet:

Afbeelding met tekst, schermopname, software, Multimediasoftware

Automatisch gegenereerde beschrijving

Figuur Data zoals die in de database komt te staan  
  
Afbeelding met tekst, schermopname, Lettertype, nummer

Automatisch gegenereerde beschrijving

Figuur Hoe de data getoond zal worden

Afbeelding met tekst, schermopname

Automatisch gegenereerde beschrijving

Figuur De code voor de transformatie

### Accessibility

Storybook had één probleem gevonden: een element binnen de EditorContent component van Tiptap had ARIA-labels. Online was hier niks over te vinden. Uiteindelijk heb ik dit opgelost door de onderstaande code toe te voegen aan mijn RichTextEditor klasse: Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

Figuur Oplossing voor accessibility probleem in Tiptap  
Afbeelding met tekst, Lettertype, schermopname

Automatisch gegenereerde beschrijving

Figuur Storybook geeft 0 violations

# Advies

In sprint 2 adviseren we om deze rich tekst editor mee te nemen in de taken waar dat nodig is.  
De rich tekst editor zelf en een voorbeeldimplementatie staat op de “rich-text-editor-test” branch.