

HADEDA

Team Chakra: Anita Ganti, Chris Maxwell & Yogendra Pawar



Elevator pitch:

The HADEDA app is a full stack web-app to automate the process of redacting sensitive data from legal documents, with the goal of making The Hague more transparent.

Concept

Harm to Ongoing Matter

- Description:
 - We built the HADEDA app to redact confidential data such as personal data and corporate information. We built a full stack - application with a REACT.js client, REST API, and MySQL database and used spaCy an open-source software library for advanced natural language processing which supports deep learning workflows connecting models trained by popular machine learning libraries. We used its features for text categorization and named entity recognition (NER). Prebuilt statistical neural network models to perform these task are available for English, German, Greek, Spanish, Portuguese, French, Italian, Dutch, Lithuanian and Norwegian.
- Motivation for development?
 - Legislation obliges the Hague to be transparent. Legal decisions and transactions of legislative bodies can only be made public if confidential data such as personal data and corporate information are erased or redacted. Erasing personal data. Currently, only 3% of available data is released. 97% is not yet released due to time consuming redaction process.
- User story
 - AS an organization who wants to be more transparent,
 - I WANT to redact sensitive data from documentation that I want to make public

Process

In this project we applied the following packages:

- "@apollo/react-hooks": "^3.1.3",
- "@testing-library/jest-dom": "^4.2.4",
- "apollo-boost": "^0.4.7",
- "apollo-link-context": "^1.0.20",
- "apollo-server-express": "^2.11.0",
- "express": "^4.17.1",
- "sequelize": "^5.9.7",
- "react": "^16.13.1",
- "react-dom": "^16.13.1",
- "react-router-dom": "^5.1.2",
- "react-scripts": "3.4.1"
- "spaCy"
- "REST-API'

Process

- Breakdown of tasks and roles:

- We all have very similar goals and work styles, so we really enjoyed working with each other.
 - We researched other React Apps to inspire our user interface and design. Anita worked on the Client side UI and logic, while Yogendra worked on leveraging the spaCy toolkit.
 - We discussed features and packages we wanted to learn to Implement to take our application to the next level and outlined the priority we had for each. Chris worked on testing the database, models, queries and seeding data. He also worked on styling, a few react components and serving our App from Heroku.
 - Documentation: Anita also worked on creating our user-story, our Readme, and our Presentation. We all created Github issues for the necessary actions, we selected specific actions we wanted to take on and assigned them to ourselves.

Process

- Challenges:
 - Using a combination of React, Hooks, MySQL, REST-API's together with spaCy (An open-source software library for advanced natural language processing which supports deep learning workflows). We also researched other API's and services including openNLP, FuzzyWuzzy, Microsoft Language Understanding Intelligent Service (LUIS), Amazon Comprehend, Microsoft Knowledge Exploration Service, IBM Watson Knowledge Studio, Stanford CoreNLP, NLTK. We also looked into API services such as TextRazor.
- Successes
 - We have come up with a minimum viable application that is intuitive and easy to navigate

D. The Areas Identified by the Government

1. Mr. Manafort's Interactions with Konstantin Kilimnik

It is accurate that after the Special Counsel shared evidence regarding Mr. Manafort's meetings and communications with Konstantin Kilimnik with him, Mr. Manafort recalled that he had – or may have had – some additional meetings or communications with Mr. Kilimnik that he had not initially remembered. The Government concludes from this that Mr. Manafort's initial responses to inquiries about his meetings and interactions with Mr. Kilimnik were lies to the OSC attorneys and investigators. [REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

Directions for Future Development

For MVP we wanted to keep the length of the string smaller. But provided the chosen database tool and processing compute power supports it, there is no limitation on the length of the document that can be processed.

These are ideas we identified to continue to build on our application even further:

1. Feature 1: Automate Publishing the redacted text to a publicly accessible database in an XML formal.
2. Feature 2: Ingest the document in the currently published format without any user interference
3. Feature 3: With time and enough training data, complete the redaction process minimizing any user input.
4. Feature 4: Increase security features and encryption in the data ingestion process.

Links

- <https://HADEDA-APP.herokuapp.com/>
- GitHub repo: anitapeppercorn/team-chakra