handling, Bottleneck for rate limiting, CORS for enabling cross-origin resource sharing, and others for file upload, Word document manipulation, and more. The backend processes LaTeX-Zip files by detecting the main file, extracting chapter order, and generating a report for download. Language detection is done to ensure the report language matches user selection. OpenAI is used for text summarization. The frontend consists of upload and download pages, following Separation of Concerns principle. Various libraries like React, React Router, and Jest are used for frontend development. The design focuses on simplicity, clear information hierarchy, and usability. The upload page allows users to input additional details for LaTeX files, while the download page facilitates file retrieval based on passed state parameters. The process involves React Router's navigation and state management. The download function creates a temporary link for file download based on specified URL, title, and format. Overall, the communication and interaction between frontend and backend facilitate seamless report generation and download for Reforge users.

VI. Evaluation and results

The chapter "Evaluation and Results" discusses the evaluation and results of a project focusing on configuration management, API usage costs, report quality, mobile usability, and comparison with existing tools. It also covers the process of generating keys for OpenAI and DeepL, along with integrating them into the application. The process of setting up and running the application locally is detailed, involving key generation and configuration settings. The costs and quality of the OpenAI and DeepL services are analyzed, along with a comparison with other tools like Chat, Copilot, Gemini, and QuillBot. Lessons learned from the project are summarized as valuable insights for future endeavors.

VII. Project outlook from Reforge

In this chapter on "Reforge Project Outlook," potential expansions and improvements are discussed, focusing on text cleanup for LaTeX documents. Suggestions include enhancing text cleanup to preserve code sections, improving handling of content like comments, automating integration of the bibliography, implementing a database for data storage and user actions tracking, considering inclusion of images in processing, adding user-defined parameters to the frontend, enabling email delivery of generated reports, and managing document generation limits through a credit system to control costs. These improvements aim to enhance user control, efficiency, and sustainability of the application.

VIII. Conclusion

The conclusion summarizes the work and reflects on achieving the defined goals. A web application was developed and successfully implemented in this bachelor thesis. It can summarize submitted documents and provide them in specific formats like IEEEtran or the OTH research report. Modern techniques for text summarization and generation, as well as multilingual support in German and English, were integrated. The application offers a platform for efficient creation of technical reports for students and teachers. It processes LaTeX-ZIP files and Word documents, and integrates tools like OpenAI and DeepL for text summarization and translation. The study shows that such a tool can reduce time and effort, but also highlights limitations like token limits and the need for human validation. The work contributes to text summarization and academic tool development. It confirms meeting project goals and functional requirements, with attention to architecture, text summarization integration, and report quality. All defined requirements have been fulfilled, including capturing document text, generating segments error-free, and supporting various output formats. The application also meets multilingual requirements and has a functional user interface. In conclusion, all goals and requirements were achieved, making the application operational for document processing and summarization.

IX. Pictures

Summary: The chapter "Images" is discussed or focused on.

X. Code extracts

Summary: The chapter is about code excerpts.

XI. Bibliography

This report was generated from the uploaded file using OpenAI. All sources must be added manually before publication. Name of the file: BT2024StrickerNatalieThesis.zip