



Engineering
& Computing

School of Computing
& Information Sciences

School of Computing & Information Sciences

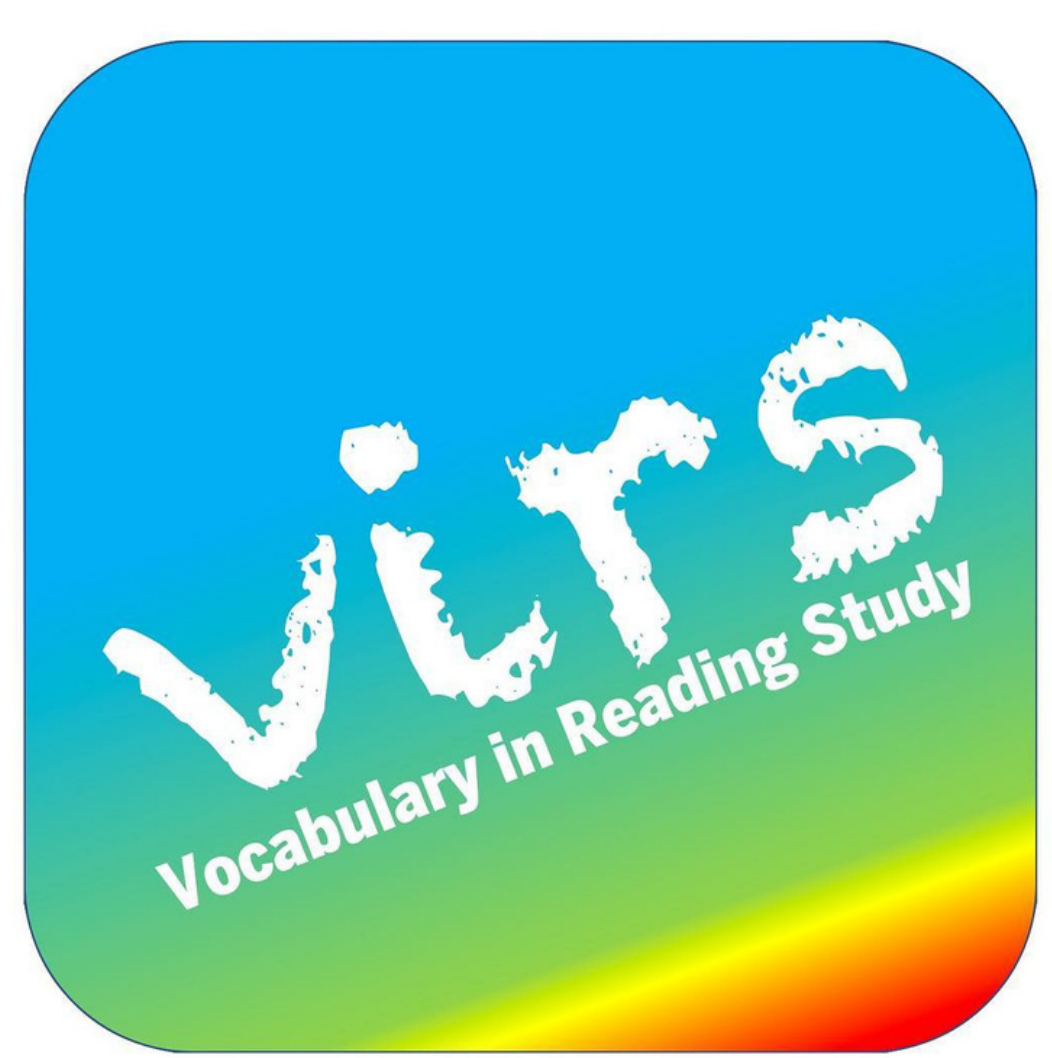
Summer 2025 Senior Design Project

VIRS - Vocabulary in Reading Study

Students: George Spanioli, Issik Drew, Joshua Lopez, Ivan Marquez Jr, Justin Robertson

Mentor: Eric Dwyer, FIU

Instructor: Dr. Masoud Sadjadi, Florida International University



Problem

1. At the beginning, the website was at a non-operational state.
2. The UI of the site was outdated.
3. Limit of up to 25 MB when uploading files.
4. Uploading files would sometimes not work or just wouldn't work with the "Enhance text" feature.

Current System

Vocabulary instruction relies on manual methods such as generic dictionaries and pre-made word lists. There is no integrated tool to ingest classroom texts, analyze word frequency or difficulty, or highlight key academic vocabulary. Teachers must individually select and assess words, and available resources do not provide data-driven support tied to specific curricula.

Requirements

Ingest and parse uploaded text files (PDF, Word, and plain text) and compute word frequency and difficulty using educational corpora. Automatically highlight key vocabulary and academic terms, and provide an intuitive interface for teachers to upload text and view results. Deliver accurate, usable, and scalable performance.

System Design

- The program/website layout:
1. Frontend: The site uses CSS, HTML5, Angular Typescript, and Bootstrap.
 2. Backend: Java Springboot and MySQL.
 3. Hosting: The Site is hosted on AWS and deployed on Elastic Beanstalk.
 4. Database: The wordlist is stored in MySQL.

Object Design

1. TextProcessor: Parses uploaded text, tokenizes words, and cleans punctuation.
2. VocabularyAnalyzer: Calculates word frequency and difficulty using corpus statistics and identifies key academic terms.
3. UserInterfaceController: Manages pages for uploading texts and displaying analysis results to teachers.
4. DatabaseManager: Handles connections to the word-frequency database and caches analysis results.

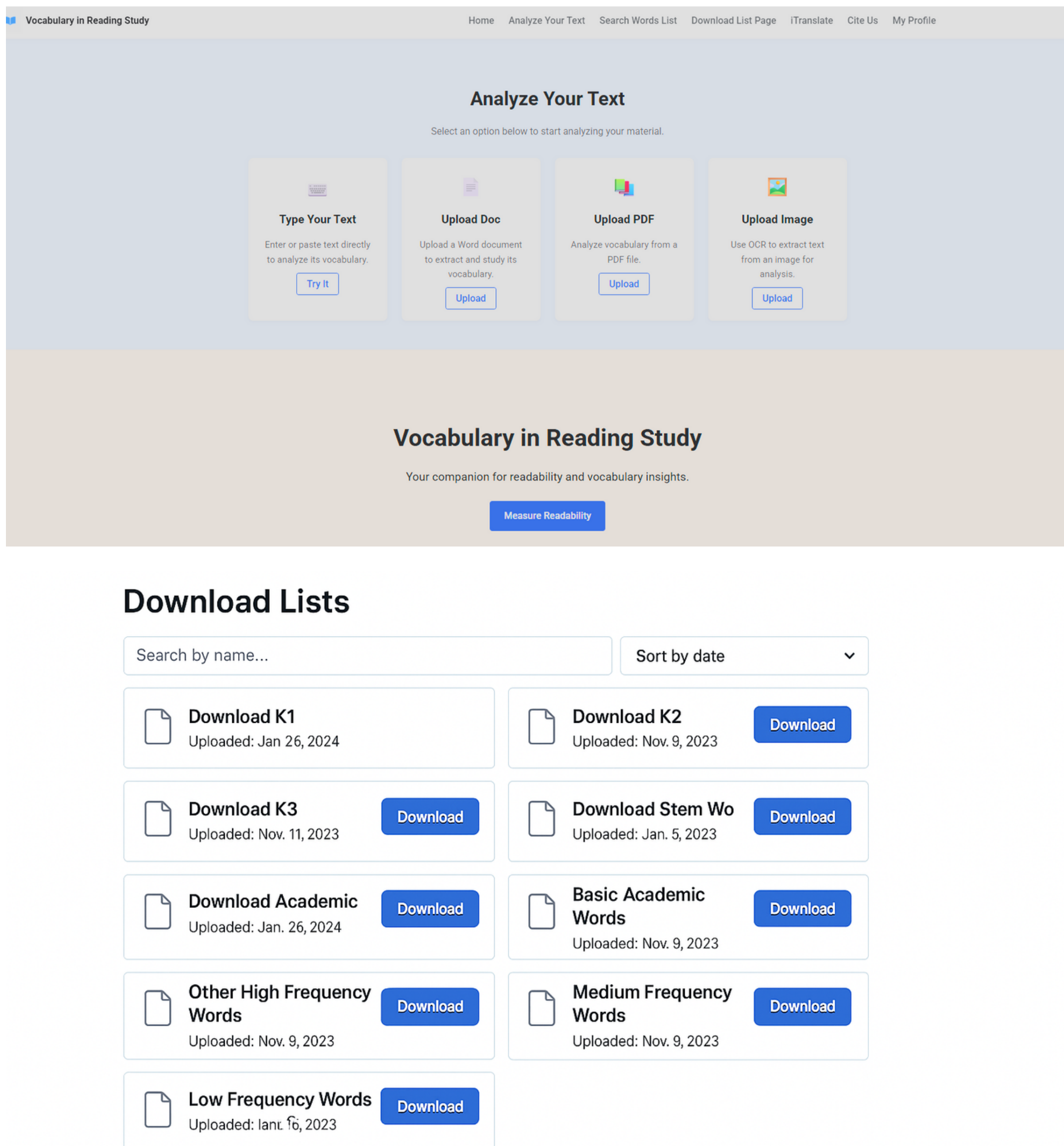
Implementation

1. Frontend built using Angular and Bootstrap, providing a responsive and user-friendly interface.
2. Backend implemented with Java Spring Boot and Python NLP libraries (spaCy, NLTK) to handle text parsing and vocabulary analysis.
3. Word frequency and difficulty data stored in a MySQL database and accessed via REST APIs.
4. Deployed on AWS Elastic Beanstalk for scalability and reliability.

Verification

1. Unit tests verify correctness of text parsing, tokenization, and vocabulary analysis algorithms.
2. Validation against established word lists and grade-level corpora confirms accuracy of frequency and difficulty metrics.
3. User testing sessions with teachers ensure results are meaningful and the interface is intuitive.
4. Performance tests measure response time and scalability with large texts and multiple users.

Screenshots



Summary

Our system ingests classroom texts, analyzes vocabulary, and highlights words by frequency and difficulty, empowering teachers to deliver targeted instruction. We implemented the tool using Angular, Spring Boot, and Python NLP libraries, achieving accurate and scalable results. Ongoing validation and teacher feedback confirm its usefulness.

Acknowledgement

The material presented in this poster is based upon the work supported by FIU. We/I thank Eric Dwyer for assistance, cooperation and mentorship that I/we received throughout this process