Control Alt Elite

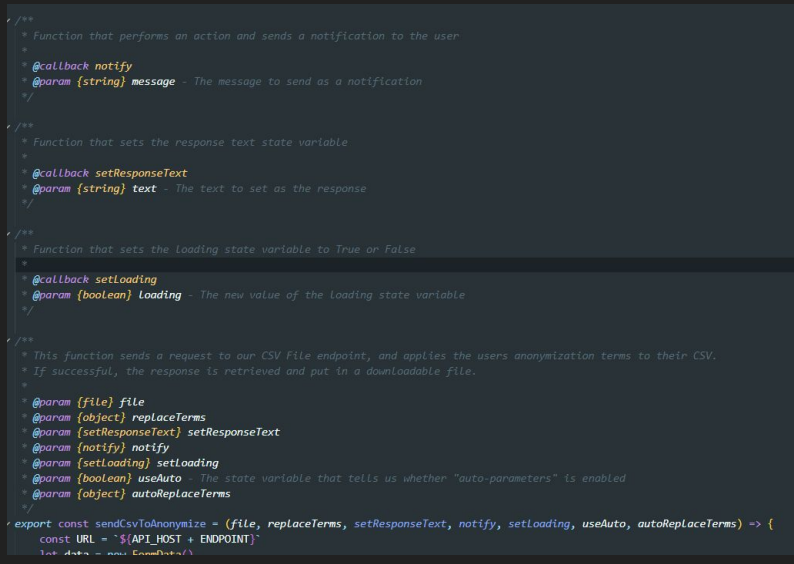
CSC301 Deliverable 3

Problem Introduction

- Partner: David (U of T CSC301 Community Partnership Program)
- “How can we facilitate storage of sensitive student data for universities?”
- Users: Professors, TAs
- Benefactors: University Students
- Solution must be flexible + account for automation

Architecture & Technical Discussion

- Tech Stack:
  - Frontend: React
  - Backend: Flask
  - DevOps: Github Actions, Google Cloud Platform, Docker
- Coding Practices:
  - Comments & Documentation
  - Descriptive variables and function names
  - Use of Github-flow (Branches, Reviewing PR’s)
  - Enforced Python linting (Black)

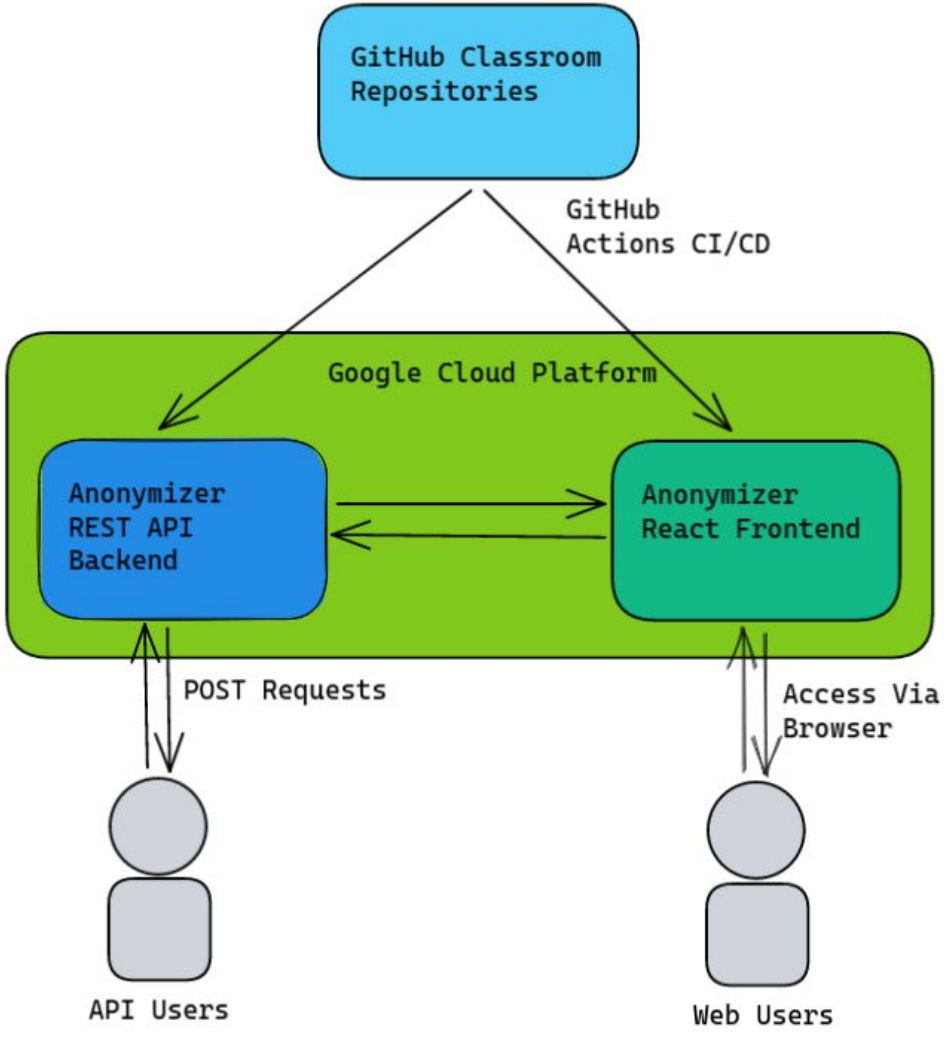


```
/**
 * Function that performs an action and sends a notification to the user
 *
 * @callback notify
 * @param {string} message - The message to send as a notification
 */

/**
 * Function that sets the response text state variable
 *
 * @callback setResponseText
 * @param {string} text - The text to set as the response
 */

/**
 * Function that sets the Loading state variable to True or False
 *
 * @callback setLoading
 * @param {boolean} loading - The new value of the loading state variable
 */

/**
 * This function sends a request to our CSV file endpoint, and applies the users anonymization terms to their CSV.
 * If successful, the response is retrieved and put in a downloadable file.
 *
 * @param {file} file
 * @param {object} replaceTerms
 * @param {setResponseText} setResponseText
 * @param {notify} notify
 * @param {setLoading} setLoading
 * @param {boolean} useAuto - The state variable that tells us whether "auto-parameters" is enabled
 * @param {object} autoReplaceTerms
 */
export const sendCsvToAnonymize = (file, replaceTerms, setResponseText, notify, setLoading, useAuto, autoReplaceTerms) => {
    const URL = `${API_HOST + ENDPOINT}`
    let data = new FormData()
```


GitHub Classroom
Repositories
GitHub
Actions CI/CD
Google Cloud Platform
Anonymizer
REST API
Backend
Anonymizer
React Frontend
POST Requests
Access Via
Browser
API Users
Web Users

Backend Logistics

Endpoint:

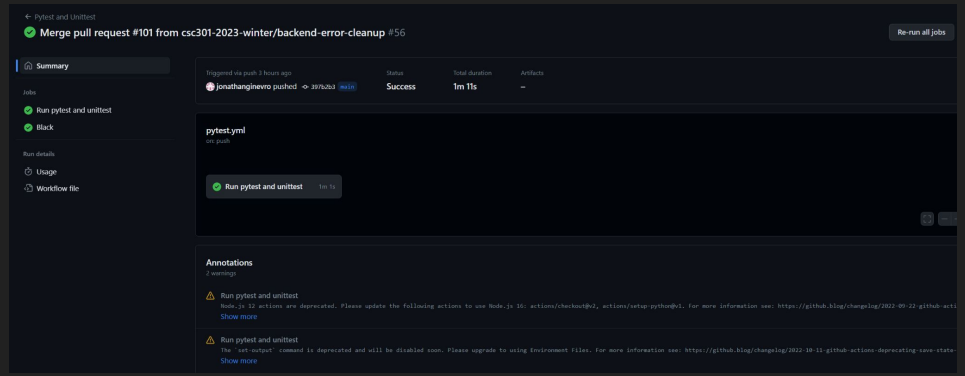
- Text replace endpoint: Anonymizes an input string
- Text File replace endpoint: Anonymizes a .txt file
- CSV File replace endpoint: Anonymized a .csv file

Implementation:

- There are two different implementations for anonymizing.
  - Auto replace feature will use NLP Huggingface API request.
  - The user specifies the specific terms and each term has all it’s occurrences manually replaced

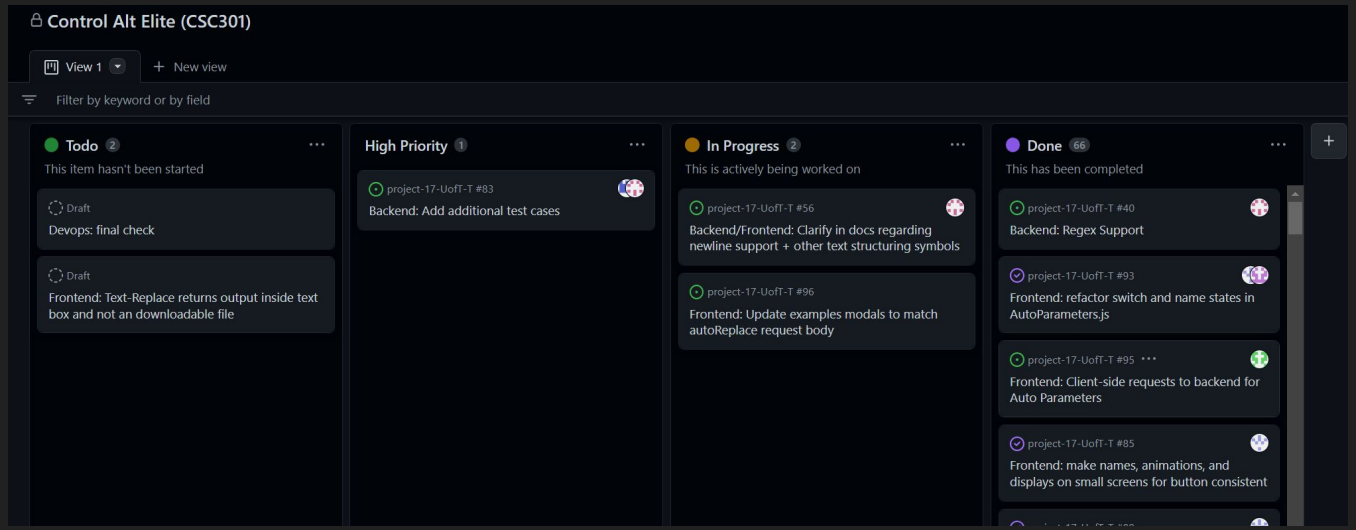
Process & Deployment

- Push-Button deployment + container management on Google Cloud via Github Actions
- Unittest written by Devops from client's perspective
- Pytest written by backend from developer's perspective
- Automated testing on PRs/commits



Process & Deployment

- Github Project: Issues + Kanban board
- Github-flow branching strategy



Access & Handoff

- Documentation + examples are provided for end users (frontend + backend)
- Cloud: Transfer service account + cloud project ownership
- Repo: Transfer GitHub repo ownership
- We will transfer the information above using a AES256 encrypted zip file, with a 32 bytes random password.

Reflection

- Importance of communication
- Taking initiative when appropriate
- Smooth and timely process for merging branches:
  - discuss
  - review
  - merge
- More emphasis on synchronous communication for subteams

Demo

https://csc301-378115-frontend-4ic67og2pa-pd.a.run.app/

Individual Contributions

Letian Cheng:

Implement the GCP pipeline including deploy, clean, service account, firewall rules. Wrote dockerfile for both frontend and backend and configure their deployment. Write client side unit test for backend.

Mehrdad Ghannad:

Developed frontend component for user interaction with the auto replace feature. Added a clear button for improved usability. Optimized page responsiveness and implemented changes for design consistency.

Jonathan Ginevro:

Implemented regular expression detection to create a multi-layered smart replacement algorithm. Incorporated the smart replacement algorithm into endpoints. Updated Swagger documents.

Rafee Rahman:

Integrated client-side requests to backend for the auto replacement feature. Added warning message before the user leaves the playground page. Added a loading spinner for submit. Disabled the CSV text area.

Nathan Hansen:

Implemented automation for backend linting + automated testing, led incorporation of D2 demo feedback, assisted in resolving backend issues found with our test suite

Edward Leung:

Implemented togglable playground menu with switches for the user to input replace terms for Huggingface API. Refactored Playground page code for readability and extendibility.

Sayna Sohrabzadeh:

Implemented the common functions for Huggingface API and helper functions. Created Postman testing collection. Incorporation of automatic name, organization, and location detection into endpoints