IT Project - Syntax Sorcerers

About

How can we ensure that the written word maintains its integrity in an era of rapid digital transformation? Driven by curiosity and the desire to make a meaningful impact, we embarked on this quest.

We are a dynamic team of third-year university students on a mission to revolutionize the way authenticity is upheld in the realm of literature. As computer science students, we recognized the significance of preserving the voice and style of authors in an increasingly AI-focused world.

Our passion for technology and literature fuels the creation of our authorship verification web app. With a blend of cutting-edge algorithms and intuitive design, we empower users to distinguish genuine authorship, unveiling the essence of each writer's unique expression within the realm of education.

Mission and vision

As Syntax Sorcerers, we are dedicated to crafting top-tier software solutions that surpass client expectations and adapt to an ever-changing digital landscape. With a foundation of collaboration, creativity, and continuous improvement, we strive to deliver excellence and champion creativity.

Our mission is to reshape the realm of authorship verification, by merging cutting-edge algorithms with our tight bond (after less than a single semester of knowing each other 😎)

Meet the team

Bunny Phetmunee
Product Owner
Dialogue/Communication with the Client
Front-end Development

Yash Patil
Scrum Master
Tracking and Delegating Tasks Through Agile
Back-end Development

Talym Myler
Developer
Front-End Development and Team Support

Antonio Felicissimo
Developer
Back-End Development and Team Support

Muhammad Usman
Developer
Full Stack Development and Team Support

Important Pages

- Project Requirements
- Authorship Front-End Design (Figma Designs)
- Authorship Front-End Design (Interactive Prototype)
- Personas
- Standup notes
- IT-Project-Syntax-Sorcerers
- GitHub - The-Syntax-Sorcerers/ava

System Structure

Git Repository Structure

Repo: GitHub - The-Syntax-Sorcerers/ava

We’ll try to be as consistent as possible. Please familiarise yourself with the repository structure below & try to follow this as closely as possible.

Note: `venv` directory is git ignored, so that folder will only exist in your local repo & it should be automatically configured by PyCharm IDE.

```
├── flaskr/
│   ├── __init__.py
│   ├── db.py
│   ├── schema.sql
│   ├── auth.py
│   ├── blog.py
│   ├── templates/
│   │   ├── base.html
│   │   ├── auth/
│   │   │   ├── login.html
│   │   │   └── register.html
│   │   └── blog/
│   │       ├── create.html
│   │       ├── index.html
│   │       └── update.html
│   └── static/
│       ├── style.css
│       └── images/
├── tests/
│   ├── conftest.py
│   ├── data.sql
│   ├── test_factory.py
│   ├── test_db.py
│   ├── test_auth.py
│   └── test_blog.py
├── venv/
├── setup.py
└── MANIFEST.in
```

CHANGE LOG

Sep 9: Front-End Transition to Flask + Vite (React)

Hey Team, just thought i'd document how to run the `flask + vite` project & some new changes. Highly recommend reading for Dev Team, it is worth the time as the reqs & `make` commands below are likely to stay for the entire duration of the project.

I will shortly merge to `dev`, so team can pull and start working on frontend.

How the app works:

- The React frontend we write (In `tsx`) is converted into static `html` & `js` by `Node`/`npm` which compiles in you `client/dist`.
- Once this static content is built, Flask uses this `dist` folder to serve only the required `js` files for a particular webpage.
- The architecture will stay consistent now for the duration of this project.

Major Changes:

- We are now running on `Flask + Vite + React + Typescript`
- Still using `Supabase` & auto-deployed on `Vercel`
- Changed name from `flaskr` to `server`
- `client` folder structure has completely changed due to the framework.
- A Github bot is setup in a way which automatically build client files (`client/dist` folder) on Github.
- DO NOT COMMIT: `client/dist` folder & your `.env`
- Deleted the html templates & css we previously had. (You can still access them on branch `pg1` (if needed))
- Deleted the previous Flask Blueprints (as they were serving incorrect templates. (You can still access them on branch `pg1` (if needed))

Tech/Stack Requirements:

- You need to have `Node` & `npm` installed from here:
- Flask dependencies will be installed using `pip` automatically on setup.
- have the `.env` present in your local (as usual)

Instructions for Backend Devs:

- Anything to do with database interaction goes in the `models.py` file.
- If the function is accessing `Supabase.Storage`, it should be created under the `Storage` python class.
- If the function is accessing `<tableName>` table (eg `User`), it should be created under the `<tableName> ` python class.
- Whenever you `render_template` at the end of a route, it requires a `template_data` (Python dict)object as a parameter in the function. This dict is then made available to the react components rendering on the page through a variable called `globalThis` (you don't need to know about this)

Instructions for Frontend Devs:

- Tutorial: Creating a new route (eg `dashboard`):
- Step1: If you want to create a new route (`dashboard`), create a folder under `src` called: `routeDashboard`

- Step2: Create 3 files under that directory, `index.html`, `dashboard.tsx`, `dashboard.css`. You can copy these from other routes and then update them accordingly.
- Step3: Add the route details in `vite.config.ts`, like this: dashboard: `resolve(root, 'routeDashboard', 'index.html')`
- Step4: `Drink Some Water (VERY IMPORTANT)` while vite-rebuilds.
- Step5: Once done, You need to add a route in server to serve this page like this: `flask.render_template('routeDashboard/index.html', template_data=template_data)`

How to Run (Final App - Assuming development is complete):

`make setup` (Only the first time)
`make run` (Starts the debug mode where new changes are tracked & updated)

How to Run (development mode - Refreshes every new file change):

For Mac,

`make setup` (Only the first time)
`make` (Starts the debug mode where new changes are tracked & updated)

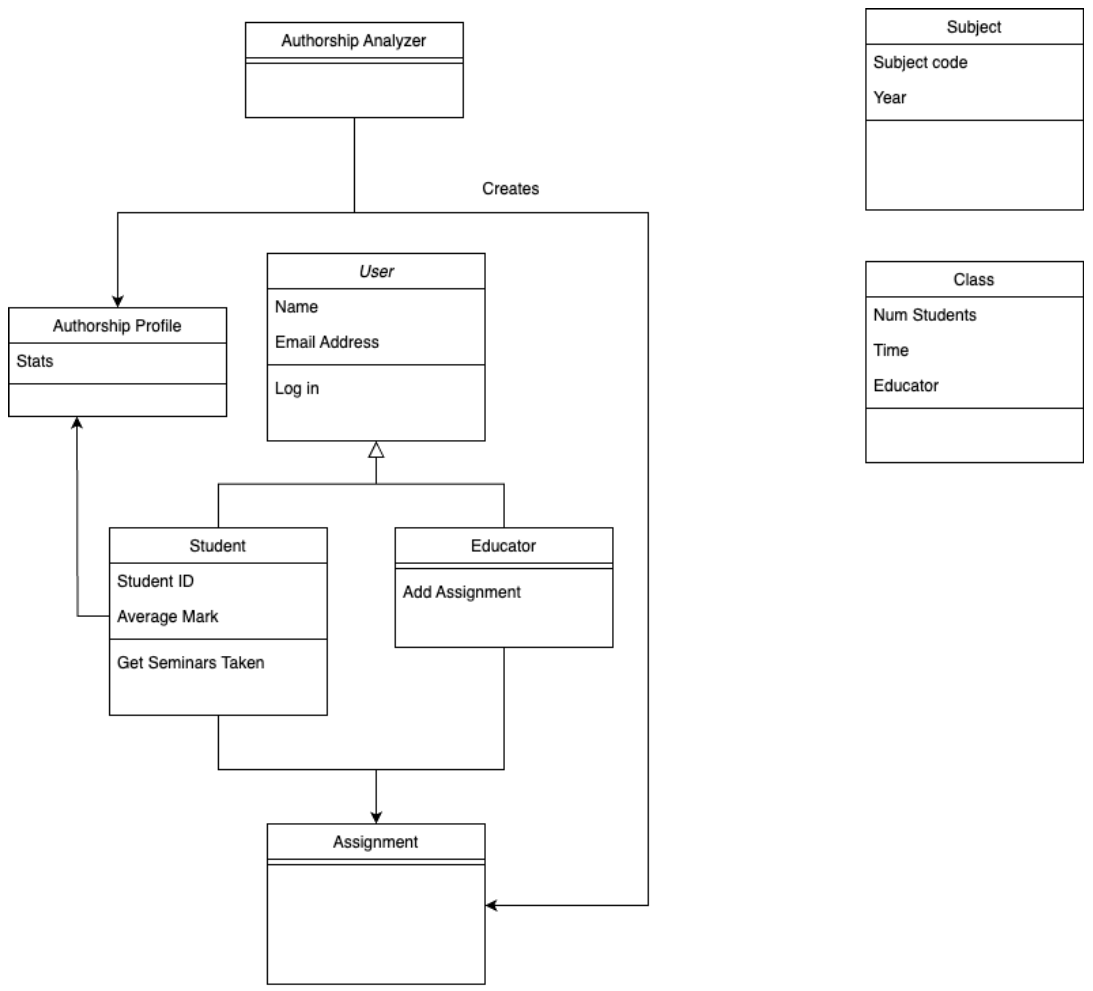
For Windows,

`make setup` (Only the first time)
`make -j2` (idk man windows is annoying, Starts the debug mode where new changes are tracked & updated)

Do not hesitate to ask any questions!!

Design Class Diagram

Our Design Class illustrates the conceptual entities and relationships we want to represent in our web app.

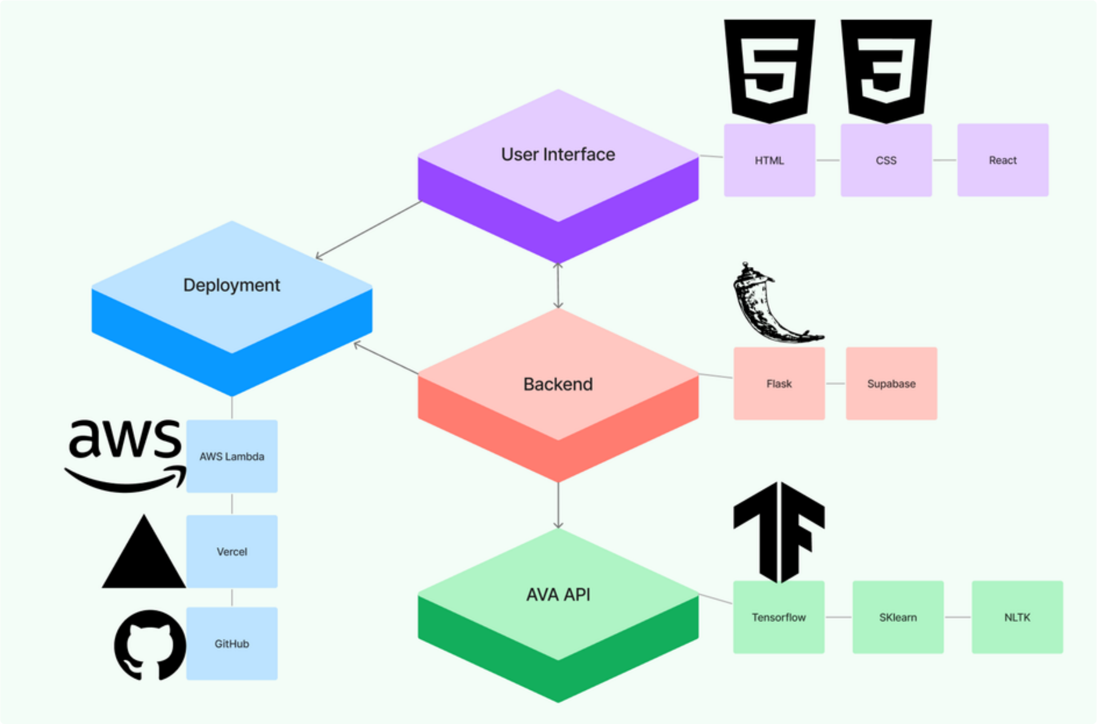


Tech Stack

AVA Tech Stack

Our current tech stack is split into four categories across 3 different layers.

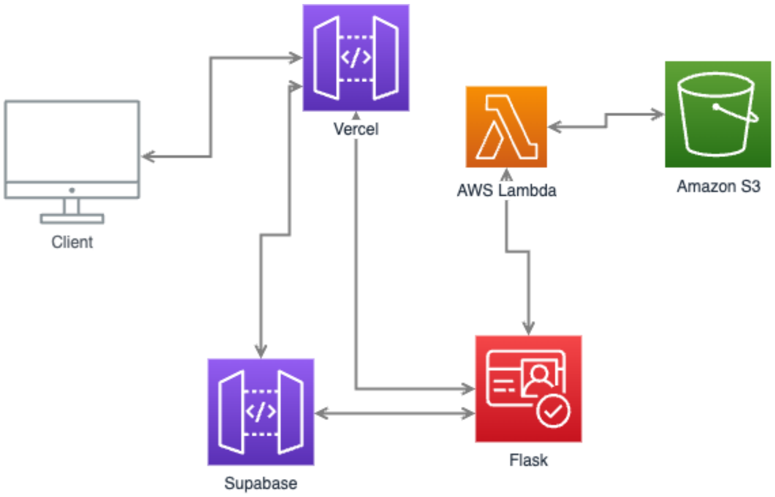
1. The top layer/client-side/user interface/front end will be written primarily using HTML, CSS, and React JS
2. Our second layer is our backend, which is divided into:
   a. our server-side functionality provided by Flask and Supabase.
   b. Hosting and deployment provided by Vercel & AWS
3. A separate API component which is the core focus of our product, the authorship verification algorithm, is already provided to us and uses Tensorflow, SKlearn, NLTK, and other Python libraries.



Architecture Diagram

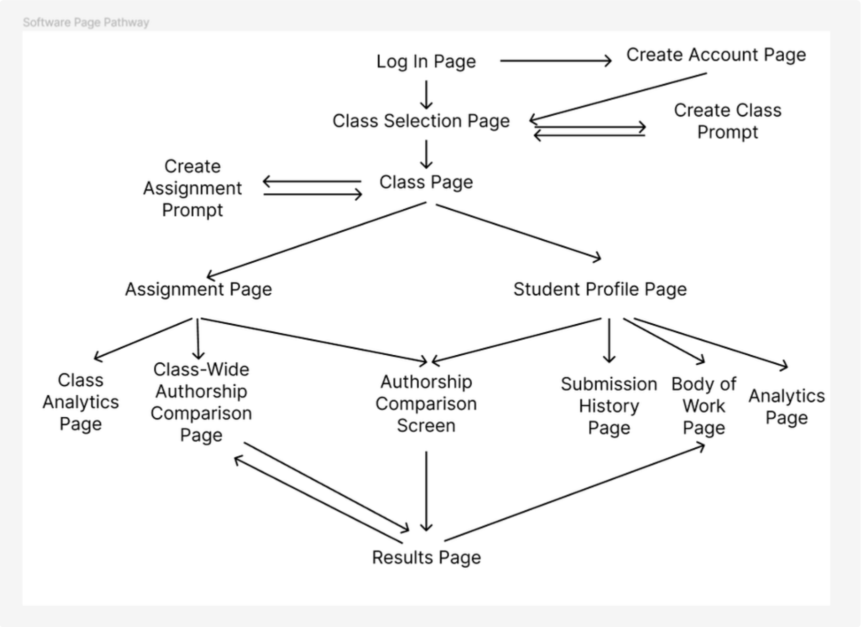
This diagram shows how data flows between the different endpoints in our software architecture.

1. The client will first log onto our app and the data will be sent through Vercel to our Flask App.
2. User data will be stored in Flask and there will be a 3-communication cycle between these three components
3. When the user submits a document, the data will be stored in Supabase and then sent to an AWS Lambda function, which will run the ML Service. The model weights for the ML model will be stored and loaded from S3.

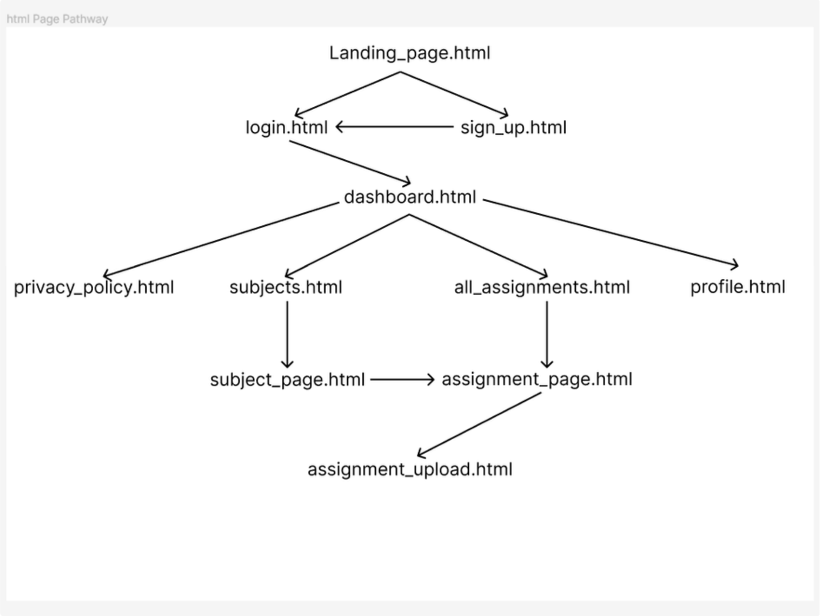


Webpage Pathways

When First designing our application, we had many pages and functionalities we initially wanted to include in our app. This diagram shows a map of how a user would navigate through our web app:



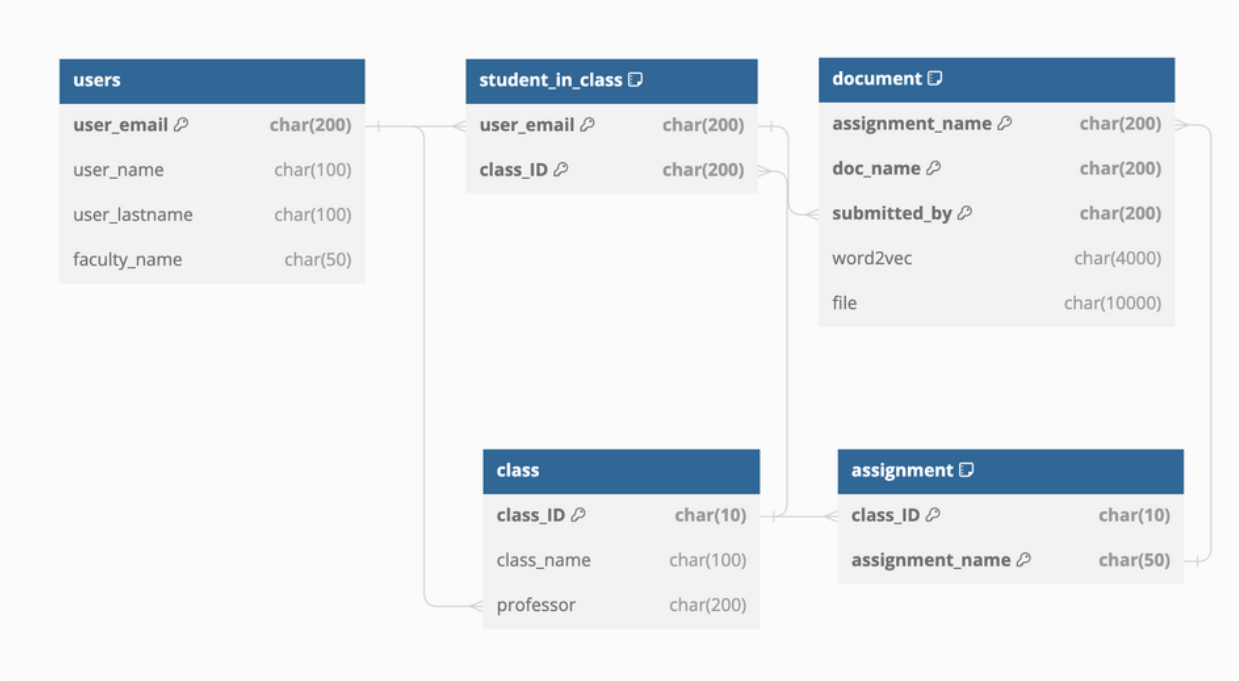
After reflecting on our first few sprints and refining the scope of our project (we were too ambitious in the beginning), these are the pages that we want to focus on to deliver a fully functioning product:



Database Diagram

This database diagram shows how will store data in our database management system (Supabase).

1. The user table will store both teachers and students, the unique identifier for each user will be their email address.
2. The student-in-class table will map each student to a class.
3. Each class will be assigned to one professor/educator to manage
4. The assignment table will contain information and details for each assignment (e.g. due date, description, possible marks awarded)
5. Each document submitted will be linked to a class, assignment, and a student, and we will have a field that stores a file location for the document and its pre-processed vector model



users	
user_email	char(200)
user_name	char(100)
user_lastname	char(100)
faculty_name	char(50)

student_in_class	
user_email	char(200)
class_ID	char(200)

document	
assignment_name	char(200)
doc_name	char(200)
submitted_by	char(200)
word2vec	char(4000)
file	char(10000)

class	
class_ID	char(10)
class_name	char(100)
professor	char(200)

assignment	
class_ID	char(10)
assignment_name	char(50)

A Free Database Designer for Developers and Analysts

API Documentation

Overview

This API documentation covers two main modules of a web application: `bp.py` and `auth.py`. The `bp.py` module handles routes related to a user dashboard and assignment uploads, while the `auth.py` module manages authentication routes for login, signup, confirmation, and logout functionalities.

`bp.py`

1. Dashboard

- **Endpoint**: `/dashboard`
- **Method**: `GET`, `POST`
- **Authentication**: Required
- **Description**: Provides access to the user's dashboard.
- **Parameters**: None
- **Returns**: Renders the `dashboard.html` template.

2. Upload

- **Endpoint**: `/upload`
- **Method**: `GET`, `POST`
- **Authentication**: Not required
- **Description**: Allows the user to upload an assignment.
- **Parameters**: None
- **Returns**:
  - `GET`: Renders the `assignment_upload.html` template.
  - `POST`: Redirects to the homepage.

`auth.py`

1. Login

- **Endpoint**: `/`
- **Method**: `GET`, `POST`
- **Authentication**: Not required
- **Description**: Provides login functionality.
- **Parameters**: None
- **Returns**:
  - `GET`: Renders the `auth/login.html` template.
  - `POST`:
    - Success: Redirects to the user's dashboard.
    - Failure: Prints error message.

2. Signup

- **Endpoint**: `/signup`
- **Method**: `GET`, `POST`
- **Authentication**: Not required
- **Description**: Provides signup functionality for new users.
- **Parameters**: None
- **Returns**:
  - `GET`: Renders the `auth/signup.html` template.
  - `POST`:
    - Success: Redirects to the login page with a successful account creation message.
    - Failure: Prints error message.

3. Confirmation

- **Endpoint**: `/complete`
- **Method**: `GET`
- **Authentication**: Not required
- **Description**: Shows an email confirmation page.
- **Parameters**: None
- **Returns**: Renders the `auth/email_confirmation.html` template.

4. Logout

- **Endpoint**: `/logout`
- **Method**: `GET`
- **Authentication**: Required
- **Description**: Provides logout functionality.
- **Parameters**: None
- **Returns**: Redirects to the login page with a successful logout message.

Tech Stack Links

Jira: IT-Project-Syntax-Sorcerers
Confluence: IT Project - Syntax Sorcerers
Github: GitHub - The-Syntax-Sorcerers/ava
Supabase: Supabase
Vercel: Dashboard – Vercel
AWS: https://ap-southeast-2.console.aws.amazon.com/console/home?region=ap-southeast-2#

Resources:

Github: GitHub - supabase-community/supabase-py: Python Client for Supabase
PAN datasets: PAN Data
React: React

feel free to add more :)

Project Requirements

🎯 Objective

Create a visual interface for an Authorship Verification Algorithm.

📊 Success metrics

Goal	Metrics
Students and Educators can create an account and log in	• Data is stored securely in a DBMS • Passwords encrypted
Educators can enroll students into classes	• Educators can view all the students currently enrolled into a class • Educators can view all current students and past students enrolled in a subject.
Educators can add assignments to a class and students can upload documents for the assignment	• Documents stored in a cloud securely • Documents can be accessed and analyzed using the analyzer

🌟 Milestones

401Client must be authenticated to access this resource.

🗒 Functional Requirements

Requirement	User Story	Importance	Jira Issue
Log in/Sign up	• As a student, I want to have access control over my account to ensure secure access to my work • As a student, I want to be able to reset my password to maintain control of my account if I forget it • As a teacher, I want to have access control over my account to ensure secure access to my structures and students' work • As a teacher, I want to be able to reset my password to maintain control of my account if I forget it	HIGH	IPSS-30 IPSS-43 IPSS-36 IPSS-37 IPSS-53 IPSS-72 IPSS-77
View and Create Classes	• As a teacher, I want to be able to organise my students into classes for easy access and ‘file'-keeping • As a teacher, I want to be able to create classes for easy organisation of students and their work • As a teacher, I want to be able to assign/dismiss students to classes for easy organisation of students • As a student, I want to be able to have assigned classes to make assignment access easy	HIGH	IPSS-31 IPSS-78
View Student Profile	• As a student, I want to be able to access my own profile to view my submission history and stylometry analytics • As a student, I want to be able to access my own submission history from my profile to easily check I submitted the correct work and the given results • As a teacher, I want to be able to access individual student profiles to troubleshoot issues	HIGH	IPSS-32 IPSS-38 IPSS-76

	they may be having with allocations and submissions - As a teacher, I want to be able to review a student's submission history from their profile for classes and assignments to be able to validate results personally - As a teacher, I want to be able to view a student's body of work from their profile to review the matching sections in the text		
View and Create Assignments	- As a teacher, I want to be able to create assignment pages to organise student work - As a teacher I want to be able to create assignment pages to allow students to submit their work - As a teacher, I want to be able to edit assignment names/due-dates/descriptions/active statuses to control information for students - As a teacher, I want to be able to view class analytics by assignment to see if the quality of my assignment specs are up to scratch - As a student, I want to have assignment pages with descriptions to easily check requirements - As a student, I want to have assignment pages with upload options to submit my work - As a student, I want to have past assignment pages to review previous submission information and results	HIGH	IPSS-34 IPSS-78
Upload documents to assignment	- As a student, I want to have assignment pages with upload options to submit my work - As a teacher I want to be able to create assignment pages to allow students to submit their work	HIGH	IPSS-35 IPSS-70

Run Authorship Verification securely and efficiently	• As a student, I want to be able to upload and submit files for assignments to validate my authorship • As a teacher, I want to be able to compare a student's work to ensure authorship • As a teacher, I want to be able to compare an ‘in doubt’ piece of work across the whole class to check if actual authorship is local (Potential Ethics Issue?) • As a teacher, I want to be able to configure the authorship comparison parameters to have finer control over the results (e.g. adjust the n-gram size, word vector size, and model type) (Potential Ethics Issue?) (Not suited for target audience?)	HIGH	IPSS-70
View results and analysis	• As a student, I want to be able to check my authorship comparison results after submitting an assignment to know the outcome of my result • As a student, I want to be able to visually view highlighted sections of my work to know what helped or hindered my results • As a teacher, I want to be able to view the comparison results to ensure authorship • As a teacher, I want to be able to visually view highlighted sections of student work to know what helped or hindered their results so I can make my own judgments in an ‘in doubt’ case	MED	IPSS-76
Body of work storage	• As a student, I want my work that has passed inspection to be added to my reference body of work to most accurately reflect my writing style • As a teacher, I want my student's work that has passed inspection to be added	MED	

	to their body of work to most accurately reflect their writing style - As a student, I want my previous reference body of work stored so I don’t have to manually add previous documents every time I make a comparison - As a student, I want my previous body of work to be secure and only accessed by parties deemed absolutely necessary to the authentication process - As a teacher, I want student’s previous reference body of work stored so I don’t have to manually add previous documents every time I make a comparison		

🗒️ Non-Functional Requirements

Requirement	User Story	Approach
Security and Privacy	As a user, I want to make sure that none of the information I provide to this service will be misused or shared without consent.	- Write a privacy policy and terms of service on the website to let users know exactly how their data will be collected, stored, and used. - When creating an account, ask users to agree to a terms of service before proceeding. - Make sure authentication and encryption services are reliable. - Make sure database storage is reliable
Efficiency	As a teacher, I want to quickly perform authorship verification on a large cohort of students.	- Use an online service that can easily scale to serve multiple requests. - Leverage GPU to speed up inference if possible
Adaptability	As a student/teacher, I should be able to rely on the output of the model as my writing style changes/as more data comes in. As a student/teacher, I should be able to trust that the model works	- The model should periodically fine tuned on new data so that it can adapt to new information and patterns. - Make sure that the model has adequate regularization to prevent overfitting

	for different types of papers/documents.	• Extensively test and evaluate whether the model is reliable using a variety of datasets.
User Feedback and Monitoring	As a user, I should be able to override or report incorrect predictions made by the model.	• Consistently ask for user feedback and allow users to report any clear mistakes made by the model. • Use feedback to further refine the model through Reinforcemnt Learning from Human Feedback (RLHF)

🎨 User interaction and design

Use Cases and User Stories

Use Case Diagram

Personas

🤔 Assumptions

- Assumes that teachers are not also students (i.e. tutors who are also enrolled in other subjects cannot be educators?)
- Each assignment is written only by one student

⚠️ Out of Scope

- Accessibility-supported UI for blind people
- Live Writing Style Analysis as you type
- Directly showing results to students through the website
- Verifying group projects

Use Cases and User Stories

User Stories

Site Access

- As a student, I want to be able to have access control over my account to ensure secure access to my work
- As a student, I want to be able to reset my password to maintain control of my account if I forget it
- As a teacher, I want to be able to have access control over my account to ensure secure access to my structures and students' work
- As a teacher, I want to be able to reset my password to maintain control of my account if I forget it

Organising students by Class

- As a teacher, I want to be able to organise my students into classes for easy access and ‘file'-keeping
- As a teacher, I want to be able to create classes for easy organisation of students and their work
- As a teacher, I want to be able to assign/dismiss students to classes for easy organisation of students
- As a student, I want to be able to have assigned classes to make assignment access easy

Organising class work by assignments

- As a teacher, I want to be able to create assignment pages to organise student work
- As a teacher I want to be able to create assignment pages to allow students to submit their work
- As a teacher, I want to be able to edit assignment names/due-dates/descriptions/active statuses to control information for students
- As a teacher, I want to be able to view class analytics by assignment to see if the quality of my assignment specs are up to scratch
- As a student, I want to have assignment pages with descriptions to easily check requirements
- As a student, I want to have assignment pages with upload options to submit my work
- As a student, I want to have past assignment pages to review previous submission information and results

Student profiles

- As a student, I want to be able to access my own profile to view my submission history and stylometry analytics
- As a student, I want to be able to access my own submission history from my profile to easily check I submitted the correct work and the given results
- As a student, I want to be able to view my body of work from my profile to see my own records
- As a student, I want to be able to view my own analytics screen from my profile to gain a better understanding of my own writing style and see my trends over time
- As a teacher, I want to be able to access individual student profiles to troubleshoot issues they may be having with allocations and submissions
- As a teacher, I want to be able to review a student's submission history from their profile for classes and assignments to be able to validate results personally
- As a teacher, I want to be able to view a student's body of work from their profile to review the matching sections in the text
- As a teacher, I want to be able to view a student's analytics screen to check for inconsistencies in the most recently submitted work and the student’s past submissions

Submission History

Body of Work

- As a student, I want my work that has passed inspection to be added to my reference body of work to most accurately reflect my writing style
- As a teacher, I want my student's work that has passed inspection to be added to their body of work to most accurately reflect their writing style
- As a student, I want my previous reference body of work stored so I don’t have to manually add previous documents every time I make a comparison
- As a student, I want my previous body of work to be secure and only accessed by parties deemed absolutely necessary to the authentication process
- As a teacher, I want student’s previous reference body of work stored so I don’t have to manually add previous documents every time I make a comparison

Analytics

- As a student, I want to be able to view my stylometry analytics to better understand my writing style
- As a student, I want to be able to see the changes in my stylometry over time to better understand how my writing style is changing

Making authorship comparisons

- As a student, I want to be able to upload and submit files for assignments to validate my authorship
- As a teacher, I want to be able to compare a student's work to ensure authorship
- As a teacher, I want to be able to compare an ‘in doubt’ piece of work across the whole class to check if actual authorship is local (Potential Ethics Issue?)
- As a teacher, I want to be able to configure the authorship comparison parameters to have finer control over the results (e.g. adjust the n-gram size, word vector size, and model type) (Potential Ethics Issue?) (Not suited for target audience?)

View Results and Analytics

- As a student, I want to be able to check my authorship comparison results after submitting an assignment to know the outcome of my result
- As a student, I want to be able to visually view highlighted sections of my work to know what helped or hindered my results
- As a teacher, I want to be able to view the comparison results to ensure authorship
- As a teacher, I want to be able to visually view highlighted sections of student work to know what helped or hindered their results so I can make my own judgments in an ‘in doubt’ case

Personas

Educator 1

Persona name	**Patricia Welling**
User Type	Educator
Role description	University Professor
User ID	2153374

🏛 Institution

University name	University of Melbourne
University size	Large university with thousands of students
Subjects taught	History and Philosophy

👥 Demographic information

Age	30
Gender	Female
Education level	Master’s

🖊 Personal quote

“Authenticity and originality are necessary precursors to gaining true wisdom in the arts.”

📜 Biography

Goals	**Motivators**

• Verify the authenticity of student submissions. • Prevent plagiarism within her classes. • Provide accurate and quick feedback to students.	• Wants to challenge students to think for themselves • Wants to maintain the integrity of the university, so that they aren’t allowing unqualified students to get a degree • Wants to build trust between the academic staff and the students
Challenges	**Needs**
• Checking for plagiarism is time consuming • It is difficult to detect small changes in writing style • There is no existing tool that can seamlessly integrate into her current workflow	• Accuracy and precision in plagiarism detection • A secure and private platform to maintain anonymity and confidentiality to reduce subconscious biases in grading. • An efficient way of analyzing papers because she has to review hundreds of students' work.

Student 1

Persona name	**Kevin Kwon**
User Type	Student
Role description	3rd Year Bachelor of Art’s Student at Unimelb
User ID	2917163

🏛️ Institution

University name	University of Melbourne
University size	Large university with thousands of students
Subjects enrolled	Sociology and Philosophy

👥 Demographic information

Age	21

Gender	Male
Education level	High School

🖋️ Personal quote

“P’s get degrees”

📜 Biography

Goals	**Motivators**
• Graduate with minimal effort • Relax as much as he can in undergrad	• His parents want him to go to college, so he’s only doing it to make them happy.
Challenges	**Needs**
• University work keeps getting harder to keep up with • He stopped paying attention in lectures and has a lot of assignments piled up	• A system that can detect plagiarism so that students like him can learn the consequences of not putting in effort.

Student 2

Persona name	**Linda Chen**
User Type	Student
Role description	1st Year Masters Student at Unimelb
User ID	1239572

🏛️ Institution

University name	University of Melbourne

University size	Large university with thousands of students
Subjects enrolled	Law and Criminal Psychology

👥 Demographic information

Age	24
Gender	Female
Education level	Bachelor’s Degree

🖋️ Personal quote

“If you can’t even manage to write an assignment yourself for University, what makes you think you have what it takes to navigate the real world?”

📜 Biography

Goals	**Motivators**
• Ensure her assignments are plagiarism-free. • Improve her writing skills based on feedback.	• Wants to achieve personal growth • Maintain academic integrity so that she feels more confident working professionally • Fairness in assessment between herself and peers
Challenges	**Needs**
• Concerns about unintentional plagiarism. • Limited access to expensive plagiarism detection tools.	• a tool that can identify inconsistencies in long writing projects. • Get insights into her writing strengths and weaknesses.

Research Project to Product

🗒️ Steps involved in the research project code:

- Extract text from files - need to consider how we deal with different types of input files (pdf, txt, docx)
- Preprocessing - cleaning text
- Training a W2Vec model
- Text Vectorization - extracting stylistic features
- Load model weights - there are different models to choose from (SVM, SiameseNet)
  - Or we could train our own model from scratch using the data they provide, possibly modify some of the parameters
- Perform inference using the model using test/validation data

😠 Issues:

- confusing to navigate
- too many function calls
- no user interface
- input data only seems to contain txt files – what if we want to upload a pdf?
- there are several different models available

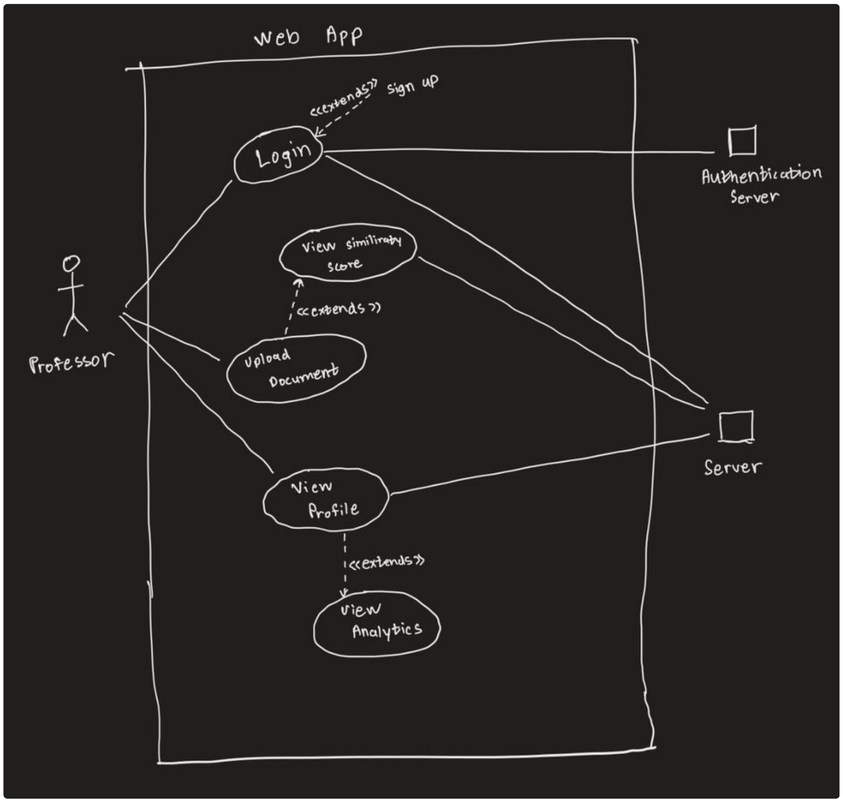
🧐 Solutions:

- Create a more streamlined program that can perform all of the steps with one function call (i.e. move functions out of jupyter notebook into a callable Python program)

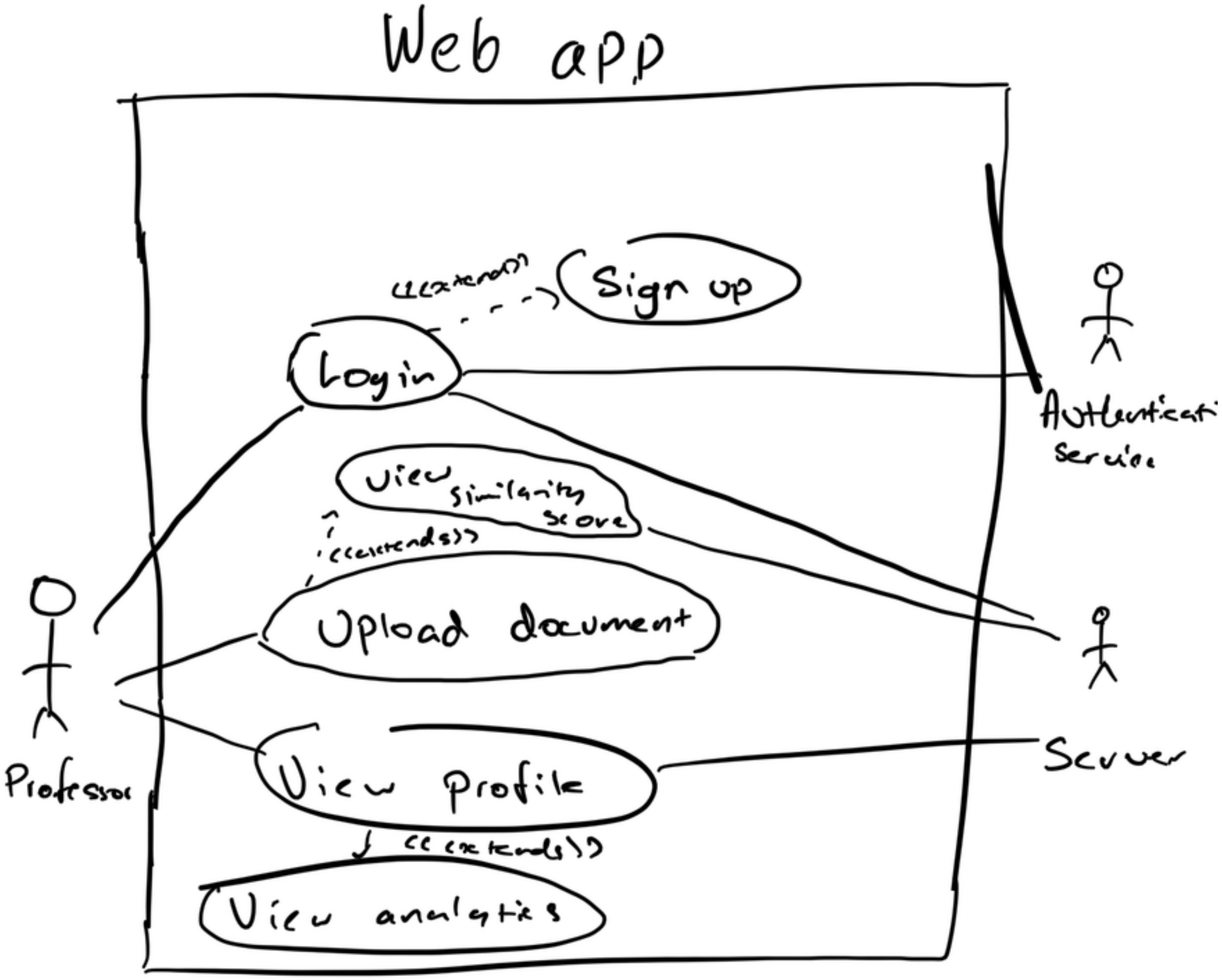
Use Case Diagram

This is a preliminary design of a use case diagram which we used to inspire our project requirements and user story planning.

Prettier Version:



Original Version:


Web app
<<extend>>
Sign up
Log in
View similarity score
<<extends>>
Upload document
View profile
<<extends>>
View analytics
Professor
Authenticati Service
Server

Standup notes

Aug 10, 2023

Date

August 10, 2023

Participants

- Full Team

Goals

- Finalise Team Role Descision
- Learn everyone's experience with web development
- Assign Stories
- Begin Sprint

Discussion topics

Item	Notes
Scrum Master Position	• Yash has filled similiar roles before and was the pseudo-scrum master for out initial ticket assignment anyway. He was also happy to do it :)
Product Owner	• Bunny has a lot of experience writing full-stack projects so will have a good understanding off all of the product and can affectively communicate our needs to the client. She was also happy to do it :)

Decisions

Scrum Master and Product Owner roles assigned to Yash and Bunny respectively - Unanimous Decision

Front-end roles assigned to Bunny and Talym, Back-end roles assigned to Antonio, Usman, and Yash - All Volunteered

Aug 11, 2023

Date

Aug 11, 2023 (In the tutorial)

Participants

- Full Team except for Talym

Goals

- Decide on the tech stack and hosting options

Discussion topics

Item	Notes
What is our full tech stack going to be?	• The backend should be written in python since the ML model will probably be called using python. • Options for the backend include: Flask vs. Django for backend. • For the front end, we're going to start with HTML/CSS for now to get a working app
What site can we use to host our web app?	• If we have a limited budget, we will probably go with a free hosting platform. • Which one is the best? Need to investigate further…

Action items

□ Look into options for the hosting platform

□ Begin learning how to use Flask, setting up the environment, and running the app locally

Decisions

- We went with Flask as our backend library because it seemed easier to learn and most of us were already familiar with Flask/Python

Aug 14, 2023

📅 Date

August 14, 2023

👥 Participants

- Full Team except Usman

🥅 Goals

- Finish vercel deployment pipeline
- Unify frontend with backend

🗣 Discussion topics

Item	Notes
How is everyone going in the current sprint?	• We have sign up, log in, Class, and assignment upload front end pages done • The backend logic is implemented
Merge Conflicts	• How are we handling merge conflicts as a team? ◦ Need to do pull requests everytime we want to add new updates to the dev branch. One or two people must review the changes before pushing. ◦ The vercel deployment should be fully functional before merging the pull request

✅ Action items

☐ Link Vercel to Git so that every time we update the dev branch or do a pull request, it should automatically deploy

☐ Try to merge all of our separate branches (Front end and back end) together and resolve conflicts as a team

Aug 18, 2023

📅 Date

August 18, 2023

👥 Participants

- Full Team

🥅 Goals

- Sprint review (sprint 1)
- Check everyone is happy with the current direction

🗣 Discussion topics

Item	Notes
Which file types for the website to accept?	• The algorithm takes .txt files so they definitely need to be accepted but we also want to accept .docx, and .pdf files as they will very likely be the most common file types created by users so will be required for a smooth adoption process • We believed the implementation of those is out of reach for the current epic so we will only consider .txt files.
How to view submitted work?	• We are discussing between using an embedded pdf file viewer and just displaying submitted text on the page • A pdf viewer would likely allow greater consistency across machines but displaying text could allow easier customisation • We believed the decision here is outside of the current epic so we will decide later
What information should we require from users to sign up?	• Email, Password, StaffID, StudentID, Faculty all considered • IDs and Faculty will likely be very different across schools so were removed for simplicity
Should we have a confirm password field?	• Most people copy and paste anyway and this increases the barrier for entry • People who forget their password can use the forgot password feature • Some people really value the feature to check they noted their password correctly •
Default light mode or dark mode?	• Light mode default is more inviting • Users should definitely have the option to toggle
Should we display the % score to the student?	• Could cause students to overanalyse and correct themselves too much when writing if they are worried about certain percentages rather than just pass or fail • Could limit a student's ability to understand their boundaries if a percentage is not shown to them • Waiting for client to confirm

Decisions

- .txt files are the priority - Unanimous Decision
- Username, Email, Password, Confirm Password will be required - In Contention (Majority Decision)
- Default light mode - In Contention (Majority Decision)

Aug 21, 2023

Date

August 21, 2023

Participants

- Full Team

Goals

- Discuss the progress of the current sprint (sprint 2)
- Check everyone is happy with the current direction
- Discuss interactions between the database and frontend

Discussion topics

Item	Notes
Should we switch to react?	• There are various blockers that we will encounter with purely html/css such as waiting for email verification before automatically logging in a new user account, embedded file viewers, displaying algorithm results, and dispalying analytics overlays on a file
Which filetypes to store in db?	• .txt will be required to show submission history • Results vectors will also be required to quickly and lightly run the algorithm • txt files could potentially be very very large so should be protected against
Should we use google auth?	• Adds to the security feel of the website • Students will very likely already have some sort of authentication app/account • Adds alterantives for users
What to prioritise for Friday with client?	• We want to highlight everything we have done so far so we need to link the assignment submission and model API call with the user login/signup pages to complete a full major use case

Decisions

- We will switch from pure html/css to react to prevent blockers from appearing in the future when implementing in-page file views and producing results from the model - This was a unanimous agreement after hearing reasons why and why not
- We should have filesize limits to prevent very expensive files from being processed - Unanimous agreement
- We are storing result vectors and txt files for each uploaded file - This was a unanimous agreement after hearing reasons why and why not
- We are considering using google auth as an alternate way to sign up for the sight

Aug 25, 2023

Date

August 25, 2023

Participants

- Full Team

Goals

- Client Meeting
- Refocus Project with Given Feedback
- Plan Next Sprint

Discussion topics

Item	Notes

Who is the project meant to be used by?	- Focus is on educators but student accounts can be created although scores should be hidden from them so just upload functionality
Are group-assignments in scope?	- It is out-of-scope
Is the algorithm adapting correctly?	- Should be working but Eduardo will check
Potential for us to get full AWS implementation with Eduardo Help	- Client is very impressed by our work and is willing to pay for any of our deployment costs!! 🙂

✅ Action items

□ Begin Migrating our current front-end to React to match our updated Figma designs

□ Deploy our ML API on AWS free tier for now

Decisions

Student functionality should be limited to seeing your statistics and uploading documents - Unanimous Decision

New navbar should be sticky - Unanimous Decision

Sep 1, 2023

Date

September 1, 2023

Participants

- Talym, Bunny, Usman

Goals

- Rectify confusion surrounding React migration

Discussion topics

Item	Notes
Should we move to react or stick with pure HTML/CSS?	- According to client, we should move to react - According to team vote, we decided on moving to react
Where should we deploy our app to keep it streamlined?	- Possible options: Heroku, Vercel, AWS, Render

✅ Action items

□ Test ML API calls to check that they work (and upload test data to Supabase) @Muhammad Usman

□ Continue setting up react reusable components (e.g. cards, navbar) @Premruthai Phetmunee

□ Set up Login & SignUp Pages with react (locally) @Talym Myler

Decisions

We should continue individually with our work despite roadblocks and waiting on other team members.

We should just get a running app, and worry about deployment later.

Sep 4, 2023

Date

September 4, 2023

Participants

- Whole Team

Goals

- Refine goals for the sprint
- Identify blockers and solutions

Discussion topics

Item	Notes

Blocker: Flask + React deployment	• We can either serve the react as static files from flask or as two separate apps • For now, we can just get write the React components locally while the production environment is still being set up.
ML Service ready to deploy	• Ava API service is ready in a docker file for deployment. • It is fully connected to supabase and uploading, prediction all work. • We need to upload to AWS but it will cost money because the docker image (2GB) exceeds the available ECR capacity (500MB). • We should get the client to pay for this
Testing	• There is a module called pytest for flask • We should integrate this into git so that everytime we do a pull request it should run testing before we can squash and merge. • We can have a .yaml workflow file that will run everytime we commit to the repo.

✅ Action items

□ Finish Setting up react @ Yash

□ Write Test Cases @ Talym

□ Extend ML Service for more functionality @ Usman

□ Get a Test Framework ready @ Antonio

□ Create UI Components for React (Login & Sign Up) @ Bunny

Decisions

We should start doing testing now and setting up Pytest while we're experiencing blocks with React

We won't get the AVA API on AWS until we have everything else integrated locally. When we're ready to deploy, we should have

Sep 8, 2023

Date

September 8, 2023

Participants

- Whole Team

Goals

- Get everyone on the same page
- Progress check-in

Discussion topics

Item	Notes
Testing	• Some Pytest cases are already done • Getting the testing setup was a struggle, took lots of time • How are we planning out our test cases ○ Functional testing ○ Test each of the routes ○ Test that the user can login properly ○ Test that the user is able to upload • Supabase has its own testing to check if we're connecting to the database correctly • Can we integrate testing with github deployment yet? → Not yet
Vite + React Setup	• React and vite has been set up by yash, and everything works in one single deployment • We need to integrate Bunny's React components into Yash's Vite-React branch • This might cause issues with testing since we won't have working links or forms.
Blockers	• The react blocker from last week is now resolved • We all had a Models of Computation Assignment due yesterday which slowed progress a bit • One blocker is that we have no money to deploy the ML service on AWS even though the docker image is ready.
Assessment Feedback	• Everything went well for our previous assessment, not much to improve on.

✅ Action items

□ Get pytest git deployment pipeline runnning

□ Integrate front-end components into Vite-React app

□ Write documentation for testing

Decisions

We need to merge vite-react onto the dev branch and delete all other branches so everyone is working with the latest updates.

We should have a separate testing plan document and a testing results document.

Sep 11, 2023

Date

September 11, 2023

Participants

- Whole Team

Goals

- Finalize vite+react update and get it live on Vercel
- Make sure everyone knows how to run it locally
- Plan the next sprint

Discussion topics

Item	Notes
Vite + React Setup	• Frontend team needs to make sure that just the client can run locally • Dev team needs to make sure that both the server and client runs
Sprint Progress	• We're doing better this sprint than last sprint • We were supposed to finish our sprint a few days ago, but we kept adding more tasks • We're moving our unfinished tasks to a new sprint
Changes to testing	• CSRF tokens – How will they affect testing? • If you don't login through our website's frontend client, you won't be able to send a POST request to the server
Sprint Planning	• What do need to do for this next sprint? • The end of this sprint should be Friday week 9, we should have a fully functioning app to show the client ◦ App should connect to ML service locally ◦ App front-end should be professional and match the figma prototype • Our testing should be done by the end of the sprint and should cover everything to do with login and signup

Action items

□ Continue with implementing testing but with the new React + Vite pages @ Talym & Antonio

□ Migrate Assignment Upload and Subject page to React

□ Design Student Profile page for React @ Usman

□ Get React Frontend connected to Supabase @ Yash

Decisions

We decided which tasks need to be done by the end of our next sprint and assigned them on Jira

Sep 18, 2023

Date

September 18, 2023

Participants

- Whole Team

Goals

- Determine current sprint progress
- Finalise timeline for week 9 progress report deliverables
- Discuss feedback from progress report 1

Discussion topics

Item	Notes
Verification Workflow	• Should documents get stored if verification upload fails? • Should the final submission be the only thing to be stored? • Student goes on page, selects files, clicks submit and then… ◦ Run verification and then send to database or ◦ Send to database then run verification
Clashing colour schemes in upload page	• Different branches have different colouring for that page
Focus for next sprint	• Teacher dashboard for comparing work with students • Make failed login/signup pretty and visually more distinct • Implement storage/logic for submitted work • Document preview • Add student interface for subjects
Do we have enough for current progress report?	• We believe so but we still want to get more features in if possilbe
Progress Report 1 Feedback	• Lost marks for not having motivational model ◦ We felt like we didn't need one as our use cases heavily covered the content of the model and the client has a very good understanding of the domain and coding in general • Besides that we scored very well and are quite pleased

✅ Action items

□ Continue with implementing testing pipeline in github with secret codes @ Antonio

□ Finalise navbar dropdown menu @ Talym

□ Continue design Student Profile page and result analysis @ Usman

□ Deploy online by Friday @ Yash

□ Create 'plus' card buttons that call create form @ Bunny

Decisions

Send to database then run verification - Unanimous (but also easiest)

Deploy online tonight rather than just locally even though it will cost $2 - Unanimous (but hard to swallow)

No need for a motivational model in this circumstance - Unanimous

Sep 22, 2023

📅 Date

September 22, 2023

👥 Participants

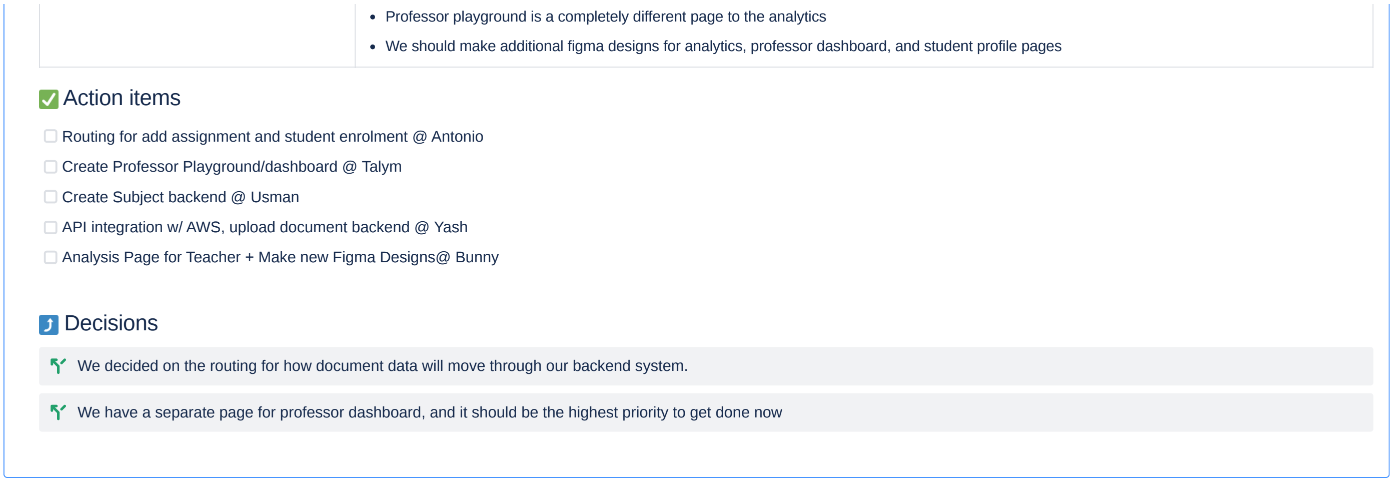
- Whole Team

🥅 Goals

- Finish up our sprint and reflect on progress
- Plan out our next sprint

🗣️ Discussion topics

Item	Notes
Sprint Planning	What we need to do next • Backend for professor pages • Document upload integration to backend • Make a dashboard for the professor to run verification for any of his students and view analysis
Test Case documentation	• Test case documentation is done • Workflow is being integrated as the react website is being done • We have 25 different test cases.
React changes	• We need to merge our branches together • Current latest changes are a bit unstable and buggy • Our website looks pretty good and clean
Analysis page	• What should the student see



	- Professor playground is a completely different page to the analytics - We should make additional figma designs for analytics, professor dashboard, and student profile pages

✅ Action items

- ☐ Routing for add assignment and student enrolment @ Antonio
- ☐ Create Professor Playground/dashboard @ Talym
- ☐ Create Subject backend @ Usman
- ☐ API integration w/ AWS, upload document backend @ Yash
- ☐ Analysis Page for Teacher + Make new Figma Designs@ Bunny

Decisions

- We decided on the routing for how document data will move through our backend system.
- We have a separate page for professor dashboard, and it should be the highest priority to get done now

Client Meeting Notes

Client Requirements Feedback

Presentation Link: AVA Project Requirements & Design

Comments:

Eduardo thinks:

A little sophisticated.

Likes the reaction feature on the uploaded documents.

Originally only for educators. It is feasible to build the user login & individual assignment uploads.

Extra feature questions:

Group assignment is Out of scope.

Additional requirement:

? Questions for Client

Question	Answer	Date Answered
How do we take input for different kids of documents (e.g. docx, pdf, txt)?		
How can we deal with documents which have multiple authors? (i.e. group assignments)	Out-of-scope	2023-08-25
Multiple models - which one do we use?		
How to handle old documents that no longer match user writing profile?	Potentially soft-ignore those old documents	2023-08-25
How to avoid making students scared to branch out when trying new writing styles?	Out-of-scope (Hide all scores from students)	2023-08-25
Will this model be applicable to students whose study involves a much broader group of writing assignments?	Based on data that we do not have access to, yes!	2023-08-25
Which device types should be prioritised?		

Project Brief Notes (Aug 04, 2023)

Bunny's Notes

Basic Idea

- Writing style analysis (like grammarly/Turnitin)
- Create an AI that will generate text based on your writing style.
- Authorship Verification Algorithm –
    - Given 1000 words of text, using text processing techniques to analyze
    - Stylometry – extract the way you write, writing stylistic features
        - how many long words, short words, how many typos, how many mistakes.
    - Create a textual profile/fingerprint for that individual
    - Given a new text, we compare to tell if the two texts are written by the same person – output a similarity score, and will determine if it is a different person given a threshold score.
    - Need existing data to create a profile in order to test against a new text input

Plagiarism:

- copying from existing sources and using them without referencing
- Someone else is writing but you submit it as your own, and plagiarism checkers (Turnitin) will not be able to to detect it.
- What kind of feedback can we provide
    - Highlight the text side by side
    - Show a percentage or scoreboard
    - Create an authorship profile ID – like a fingerprint
    - Show the ID of the closest matching individual.
    - Analytics
        - Over time, how has the writing style of one individual changed
        - For a group, how often are people plagiarizing.
        - Which words are most commonly used
- Bloom's taxonomy – six levels of types of questions
    - What's your favorite color – very simple question, no right or wrong
    - Complexity is higher for certain questions
    - Formal tone vs informal tone, we change vocabulary intentionally for different contexts
    - For highly complex tasks, we can still detect authorship identity with high accuracy

JGAAP

- Authorship attribution – with 20 people, we detect which texts were written by which person
- Heaps of different parameters – can select N-grams, which vectorization method

PanWebis

- community for authorship verification, has samples of datasets for stylometry sets

Research To Product

- Use a database system to store information the writing styles about all sorts of different individuals.
- Deploy the AI online – send data to server and send output back
- Detach hard coded input and bring into an visual interface
- Access control – user verification, authentication, login through Facebook or Google, data security

- Feedback and Analytics – can we show charts or graphs, tables which show information about the text analysis that is easy for users to read and understand.
- Assume purely text based
- Stakeholders = educators, academics
- Include handwritten text analysis
- What kind of input should we take – files, plaintext, websites
- Can create a new person based on new texts and test new tests against existing profiles.

Antonio's Notes

Yash's Notes

Client Description

Authorship Verification

Context: Papers with Code - Authorship Verification

Stylometric Inquiry.

- Reverse Engineering ChatGPT prompts using complex query. (include example of student text to emulate)
- Benchmark the GPT answer’s features against the base paragraph features to form a student profile.
- Assess for a particular person profile, if a text is written using a Pre-Trained model.
- Styrommetry. Provided Feature-Algorithms.
- Track Vocabulary changes for a person profile.
- Restrict authentication to @unimelb domains.
- Targeted towards uni professors/assessors. Text style is academic writing. Text-based.
- Doesn’t need to be openai based?
-
-
- Interface Description:
  - Once a text is uploaded. It is benchmarked against a selected student profile.
  - The algorithm should be trained against the text.
  - Used for assessing further profiling.
- Functional Requirements:
  - Feature to upload student texts for the algorithm to use.
  - Feature to benchmark a new text against a student’s profile.

Talym's Notes

Initial Prompt:

- Turnitin can only detect generated content
- Other services can generate authentic original content
- A system to verify authorship through unique author styles could combat this

Basic Idea:

- Process multiple provided text documents and generate a variety of features to generate style metrics for the individual user (Features such as vocabulary, short/long word usage, typos etc)
- Then stylometry to determine a ‘fingerprint’ for each author
- Then compare new texts to the generated profile using cosine similarity and using either a static or algorithmic threshold to determine whether that new text is at risk of being written by another author or if it should be included in the authors body of work

Project Requirments:

- A visual UI for non-tech users must be implemented
  - Interface for building initial profile
  - Interface for comparing new text
- From there direction is open-ended, could explore access control, analytics, visual overlay/representation of features on the page contributing most to a low score for feedback etc
- Scope is self-defined
- Target audience is higher education educators

Usman's Notes

Blueprint:

- Stylometry: Analyze writing style elements like word usage, typos, and unique stylistic features to establish textual 'fingerprints'.
- Comparative Analysis: Introduce new texts to established profiles (profile keeps updating - past texts may no longer valid. So should remove them?), resulting in a similarity score. Thresholds help determine potential author matches.
- Feedback Mechanisms: side-by-side text comparison, providing similarity percentage and showcasing closest matching profile **ID (using unique student ID. - primary key in record tables)**.

Implementation Details:

- Data Acquisition: Accumulate diverse text documents to create distinctive style metrics (e.g., vocabulary patterns, typo tendencies).
- Tech Integration: reverse engineering chatgpt. Systems like JGAAP and PanWebis can offer foundational methodologies.
- UI Components: Design a visual interface tailored for educators and academicians. Ensuring ease of profile creation and subsequent text comparison. Non technical person able to understand the use of software (such as academic from Faculty of arts).
  - may limits access between faculties.
- Access & Authentication: Implement rigorous access controls, perhaps restricting authentication to specific domains like @unimelb, ensuring heightened data security. (security and ethics - need to write report during project conclusion)
- Varied Functionalities: Cater to a range of input types (plaintext, pdf, docs)
- Target Audience: Primarily educators and academicians.
- The system's overarching goal is to effectively differentiate genuine student work from AI-generated or plagiarized content.

Retrospectives

Add Retrospective

Sprint 1 - Retrospective

📋 Overview

Reflect on past work and identify opportunities for improvement by following the instructions for the Retrospective Play.

Date	18/08/23
Team	Syntax-Sorcerers
Participants	Full Team

💭 Retrospective

Add your Start doing, Stop doing, and Keep doing items to the table below. We'll use these to talk about how we can improve our process going forward.

Start doing	Stop doing	Keep doing
• Increased code review requirments for pull requests	• Neglecting documentation of work done and plans designed	• Timely delivery of our designated stories and communicating when this isn’t possible
• Experimentation with tools and how we can/can’t use them within our project	• Overloading ourselves	• Helping each other through learning all of these new processes/languages/libraries etc
• Coding together in pairs/groups	• Making big decisions without involving the rest of the team	• Learning, growing, and sharing our skills with eachother

✅ Action items

□

Sprint 2 - Retrospective

📋 Overview

Reflect on past work and identify opportunities for improvement by following the instructions for the Retrospective Play.

Date	25/08/2023
Team	Syntax-Sorcerers
Participants	Full Team

💭 Retrospective

Add your Start doing, Stop doing, and Keep doing items to the table below. We'll use these to talk about how we can improve our process going forward.

Start doing	**Stop doing**	**Keep doing**
• Increased communication around minor tasks to prevent code overlap	• Extra work outside of designated tickets	• An extra weekly meeting to gauge progress and work together to overcome blockers
• Checking pull requests with increased care to help share knowledge	• Excessively long/late meetings to ensure everyone is engaged and discussion is impactful	• Voicing your thoughts and opinions on stories and epics to help shape the overall project
• Testing your changes outside of your local environment as well	• Huge leaps in progress without proper team consultation	• Documenting your work and associated visual diagrams on confluence to communicate your progress

✅ Action items

□

Sprint 3 - Retrospective

📋 Overview

Reflect on past work and identify opportunities for improvement by following the instructions for the Retrospective Play.

Date	Sep 11, 2023
Team	Syntax Sorcerers
Participants	Full Team

💭 Retrospective

Start doing	**Stop doing**	**Keep doing**
• Working in a pair or small groups. • If there is one task that is too big for one person to handle, then we should have two people working on it.	• Major transitions to different tools. If we want to make a big change that will affect everyone else’s workflow, we need to bring it up with the team first and make changes in small iterations.	• Good work momentum!! We made a lot of progress and completed many tickets.
• Break large tasks down into smaller ones. Big pull-requests create too many merge conflicts to resolve.	• One person taking on too many responsibilities. The work should be divided equally. When there are blockers for that one person, everyone else will need to wait. It’s more efficient to just delegate work evenly amongst the team.	• Reaching out and calling each other to work out issues we face individually.

✅ Action items

☑ Fix issues that were caused as a result of migrating to Vite-React.

☑ Start delegating smaller tasks to everyone on the team.

Sprint 4 - Retrospective

📋 Overview

Reflect on past work and identify opportunities for improvement by following the instructions for the Retrospective Play.

Date	Sep 22, 2023
Team	Syntax Sorcerers
Participants	Full Team

💭 Retrospective

Start doing	**Stop doing**	**Keep doing**
• Communicate more with client through Slack or email because we can’t always expect him to always show up to our tutorials.	• Creating huge pull requests that cause a lot of merge conflicts for everyone else and disrupt progress.	• Everyone attends and actively participates in all of the meetings.
• Do more small pull requests and code reviews so that we don’t make changes that interfere with other teammates' code.	• Not discussing with the team before making important changes to the code (i.e. changing folder structure), causing everyone to have a lot of issues with merging.	• Big decisions are taken to a majority vote, and we take everyone’s input into consideration.
• Comment more and document important changes to code so that we can better resolve merge conflicts.	• Repetitive and inefficient code, instead create more reusable classes and components to streamline our front-end.	• Misunderstandings are quickly cleared up. We always make sure everyone is on the same page.

✅ Action items

☐ Do more pair programming.

☐ Start writing more concise PRs & do frequent reviews.

☐ Email or message the client on Slack while he’s away to make sure our product still matches what he expects.

Testing

We use Pytest to run all of our testcases and use git (CI/CD) workflows to be able to run these tests when they are pushed.

Scenario 1: User attempts to log in with valid credentials

User Story	Given	When	Then
As a student, I want to have access control over my account to ensure secure access to my work	User is not logged in.	User submits a POST request to '/login' with valid credentials.	Server responds with status code 200. User is redirected to '/dashboard' and gains access to the dashboard.
As a student, I want to be able to reset my password to maintain control of my account if I forget it	User is not logged in.	User submits a POST request to '/login' with valid credentials.	Server responds with status code 200. User is redirected to '/dashboard' and gains access to the dashboard.

Scenario 2: User attempts to log in with non-existent username

User Story	Given	When	Then
As a student, I want to have access control over my account to ensure secure access to my work	User is not logged in.	User submits a POST request to '/login' with a non-existent username.	Server responds with status code 200. User remains on the same page ('/'). User is not logged in.
As a student, I want to be able to reset my password to maintain control of my account if I forget it	User is not logged in.	User submits a POST request to '/login' with a non-existent username.	Server responds with status code 200. User remains on the same page ('/'). User is not logged in.

Scenario 3: User attempts to log in with incorrect password

User Story	Given	When	Then
As a student, I want to have access control over my account to ensure secure access to my work	User is not logged in.	User submits a POST request to '/login' with an incorrect password.	Server responds with status code 200. User remains on the same page ('/'). User is not logged in.
As a student, I want to be able to reset my password to maintain control of my account if I forget it	User is not logged in.	User submits a POST request to '/login' with an incorrect password.	Server responds with status code 200. User remains on the same page ('/'). User is not logged in.

Scenario 4: User submits sign-up form with no changes

User Story	Given	When	Then

As a student, I want to have access control over my account to ensure secure access to my work	None.	User submits sign-up form.	Server responds with status code 200, indicating successful sign-up. User is redirected to login page.
As a student, I want to be able to reset my password to maintain control of my account if I forget it	None.	User submits sign-up form.	Server responds with status code 200, indicating successful sign-up. User is redirected to login page.

Scenario 5: User submits sign-up form with empty name field

User Story	Given	When	Then
As a teacher, I want to have access control over my account to ensure secure access to my structures and students' work	Name field is empty.	User submits sign-up form.	Server responds with status code 200, indicating successful sign-up. User is redirected to login page.
As a student, I want to have access control over my account to ensure secure access to my work	Name field is empty.	User submits sign-up form.	Server responds with status code 200, indicating successful sign-up. User is redirected to login page.

Scenario 6: User submits sign-up form with empty email field

User Story	Given	When	Then
As a teacher, I want to have access control over my account to ensure secure access to my structures and students' work	Email field is empty.	User submits sign-up form.	Server responds with status code 500, indicating a server error.
As a student, I want to have access control over my account to ensure secure access to my work	Email field is empty.	User submits sign-up form.	Server responds with status code 500, indicating a server error.

Scenario 7: User submits sign-up form with empty password field

User Story	Given	When	Then
As a teacher, I want to have access control over my account to ensure secure access to my structures and students' work	Password field is empty.	User submits sign-up form.	Server responds with status code 200, indicating successful sign-up. User is redirected to login page.
As a student, I want to have access control over my account to ensure secure access to my work	Password field is empty.	User submits sign-up form.	Server responds with status code 200, indicating successful sign-up. User is redirected to login page.

Scenario 8: User submits sign-up form with empty confirm password field

User Story	Given	When	Then
As a teacher, I want to have access control over my account to ensure secure access to my structures and students' work	Confirm password field is empty.	User submits sign-up form.	Server responds with status code 200, indicating successful sign-up. User is redirected to login page.
As a student, I want to have access control over my account to ensure secure access to my work	Confirm password field is empty.	User submits sign-up form.	Server responds with status code 200, indicating successful sign-up. User is redirected to login page.

Scenario 9: User submits sign-up form with empty email and name fields

User Story	Given	When	Then
As a teacher, I want to have access control over my account to ensure secure access to my structures and students' work	Email and Name fields are empty.	User submits sign-up form.	Server responds with status code 500, indicating a server error.
As a student, I want to have access control over my account to ensure secure access to my work	Email and Name fields are empty.	User submits sign-up form.	Server responds with status code 500, indicating a server error.

Scenario 10: User is logged in and accesses the '/logout' route

User Story	Given	When	Then
As a user, I want to be able to log out securely to protect my account	User is logged in.	User sends a GET request to '/logout'.	Server responds with status code 302 and location '/?next=%2Flogout', indicating an attempt to redirect to the login page.
As a user, I want to be able to log out securely to protect my account	User is logged in.	User follows the redirection.	Server responds with status code 200, indicating successful redirection. The user is on the login page.

User Story	Given	When	Then
N/A	An assignment object exists.	A new assignment is created with a due_datetime.	Check that the assignment has an id, a subject_id, a name, a due date, and a due_time.

Progress Checklists

Create from template

Progress Checklist 1

Date	03/09/23

Process

- ☑ Adherence to Agile Ceremonies
- ☑ Team Structure
- ☑ Team Internal Communication
- ☑ Team Decision Making
- ☑ Communication with Client
- ☑ Tools Used

Artefacts

- ☑ Requirements
- ☑ Front-end Design: Figma Designs | Interactive Prototype
- ☑ Architectural Design
- ☑ Coding: Flask App | ML Service API
- □ Testing
- ☑ Deploying

Progress Checklist 2

Date	24/09/23

Process

- ☑ Adherence to Agile Ceremonies
- ☑ Team Structure
- ☑ Team Internal Communication
- ☑ Team Decision Making
- ☑ Communication with Client
- ☑ Tools Used

Artefacts

- ☑ Requirements
- ☑ Front-end Design: Figma Designs | Interactive Prototype
- ☑ Architectural Design
- ☑ Coding: Flask App | ML Service API
- ☑ Testing
- ☑ Deploying