

## **Description**

We will be making a site with the primary function of allowing students to see important information about their teachers—most importantly, emails, course syllabi, and free periods/office hours. There will be a student login, which uses google sign-in to create an account (must be done with a Stuy email to ensure privacy from those who do not attend the school). There will also be a separate teacher portal, which uses a google sign-in also and makes sure that the email used is in a prewritten list of faculty emails. From the student side, you will be able to search for teachers and “star” their information, to be added to a student’s “saved teachers.” From the teacher's side, once you log in through the portal, you can mark which periods you are busy and free (because this does not always coincide with when one teaches class, i.e. lunch periods/work periods). The site will be set up so the teacher will also be prompted to enter the email they are most reachable at and their course syllabus if they have one.

Another important aspect of our project that was less relevant in other projects before is the site's security. If we have actual users logging in and submitting important data, we will need to implement stronger security measures like hashing (and possibly also salting and peppering), as well as string sanitation to protect against SQL injection attacks.

If we have time, we also are considering implementing some kind of handwriting-to-text converter as a stretch goal. Many teachers hand write lesson plans, so it might be helpful to have a place to upload screenshots of handwritten notes and have them converted to editable text. For this, we would use an AI library in python called PyWhatKit, which has training data that learns to take handwriting and convert it to typed text.

## **Ted Talk/New Learnination Incorporation**

For this project, we will be using Sass to create our CSS files. We will be using the theory/lesson on SQLite injections to prevent them from happening to our site. Our database will be SQLite since we all have the most experience using it, and we can use a relational database to implement the associations between teachers, students, and schedules that we plan to have. Lastly, we may potentially incorporate AI and neural networks into our project.

## **Steps Towards Feasibility**

A large part of this project is getting teachers to input their schedules onto this new platform. While we are still developing the bulk of the site, we will just create our own teacher accounts (we can add a fake email and password to the list of teachers that are allowed to log in temporarily as a tester account). When we finish the actual site, every group member will ask a few teachers to try it out. In the future, Dr. Haber could help reach out to more teachers. We will

prioritize the documentation of this project in order to allow future students to continue improving the platform.

## **Program Components**

### *Backend*

#### `__init__.py`

- App initialization

#### `routes.py`

- Contains all website routes
- Handles all connections between various Python modules, requests, and frontend

#### `database.py`

- Connections to SQLite3 database
- Reading and writing authentication information
- Reading and writing teacher data

#### `teacher.py`

- Handles teacher profile information and display (syllabi, office hours, etc.)

#### `student.py`

- Handles student profile information and display (saved teachers)

#### `auth.py`

- Authentication for student and teacher logins
- Sign in with Google
- Teacher whitelisting
- Password hashing + salt/pepper
- Methods:
  - `gen_login` (create a user from Sign in with Google data)
  - `hash_password`

#### `search.py`

- Implements the teacher search function

#### `handwriting.py`

- Implements the handwriting OCR function

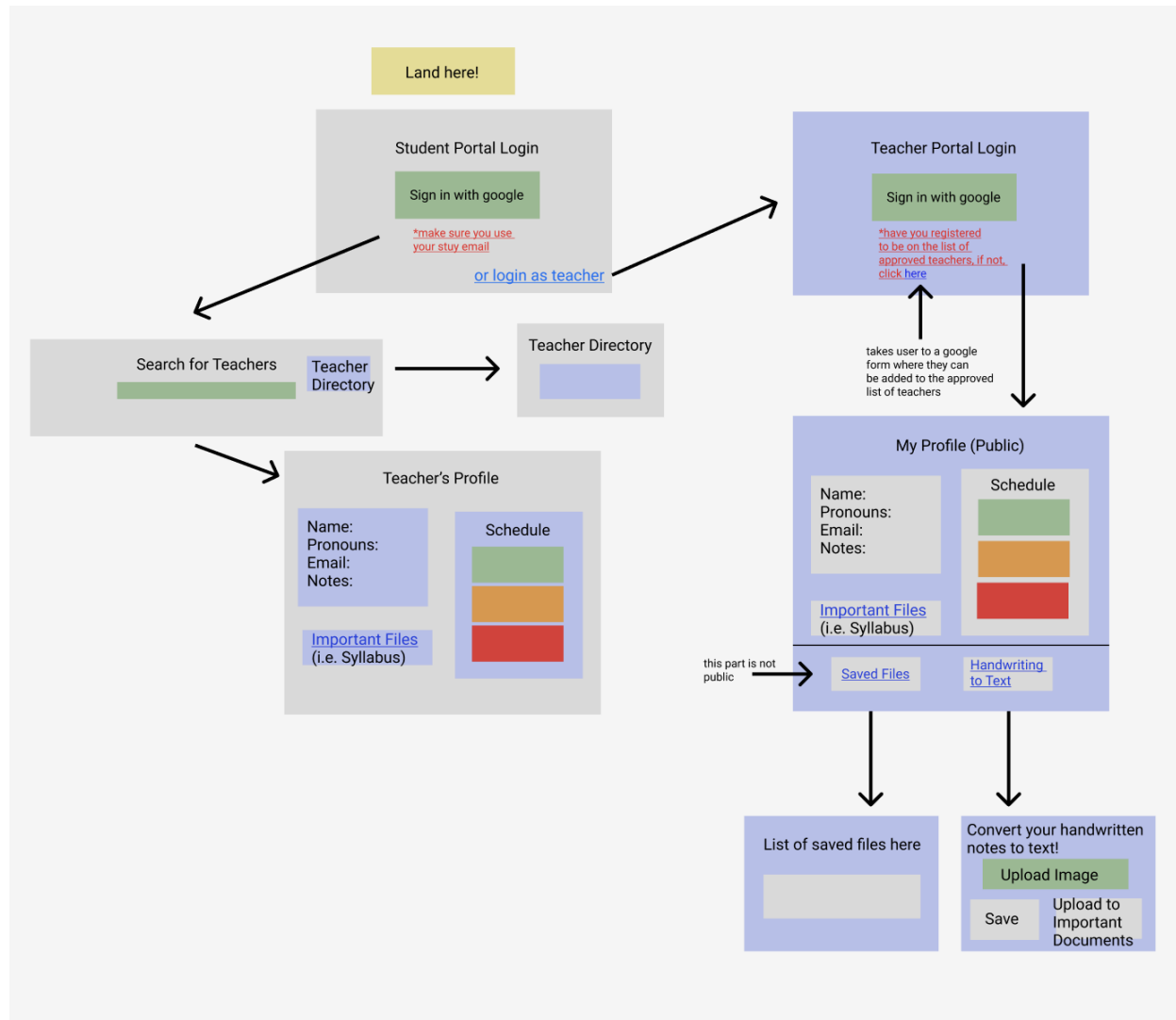
### *Frontend*

- JS components of Sign in with Google

### *Styling*

- We will use Bootstrap to style, as well as our own Sass CSS for features not included in Bootstrap
- Features in various HTML files for our pages:
  - **gen\_login.html** (can't create account— you have to log in with your Stuy email)
  - **search.html**
    - Has a search bar for teachers (you don't have to type in the complete name to search) which shows linkable results to get to each teacher's page
    - Also has a button on the side where you can access a directory of all of the teachers to find the one you are looking for
  - **directory.html**—this page just contains a directory with all of the teacher's names
  - **info.html**—this page will contain the actual teacher information
    - Name/pronouns, email, notes, and schedule. There will also be an “important files” link that will take you to a page with the syllabus first and any other important files that the teacher can add.
  - **teacher\_login.html** (teachers log in with their Stuy emails—if they want to use the portal right now, we should get them to fill out a form that goes to Google spreadsheets so we can create a list of “approved” emails... otherwise I don't know how we would authorize it)
  - **teacher\_profile.html** (contains an editable profile where teachers can fill in the information mentioned under info.html).
    - Contains a link to a handwriting-to-text converter
    - Contains a link to see private “saved files”
  - **saved.html**—contains all of a teacher's saved files

## Design on Figma



## Database

### Student

- `_id`: int
- `email`: str

### Teacher

- `_id`: int
- `name`: str
- `email`: str
- `pronouns`: str
- `title`: str (ie. Mr. Ms. Mrs. Mx. Dr.)
- `schedule`: schedules foreign key
- `period notes`: str array

- Annotations about each period for students
- files: files foreign key

### *Schedules*

- `_id`: int
- teacher: teacher foreign key
- column for each pd:
  - [p1: “free”, p2: “prep”, p3: “rm 103”, ...]
  - Options might be: free, prep, room #, not in building, etc.
- column for notes for each pd:
  - Annotations about each period for students

### *Files*

- `_id`: int
- teacher: teacher foreign key
- file name: str
- file: blob

### *Starred Teachers*

- `_int`
- teacher: teacher foreign key
- student: student foreign key

### **Tasks**

#### Andrew

- Database stuff
- Deployment
- Login
- Viewing saved files

#### Chris

- Database stuff
- AI
- Teacher directory list
- Deployment

#### Eliza

- CSS/Sass/styling
- Auth/Login
- View teacher’s profiles

Ella

- HTML/JS
- Helping out w styling
- Teacher profile (main page)

Lucas

- AI
- Flask routing
- Search for teachers