

FINAL PROJECT

Pd 9 | DaMisfits

Giorgio Vidali, Kevin Li, Cynthia Cheng, Jacqueline Xu

WHAT DOES IT DO?

Our project will help large classrooms facilitate attendance processes: allowing section leaders to take attendance and excuse absences, teachers to monitor overall absence and lateness trends, and students to check their own records and find important announcements and rehearsal dates.

COMPONENTS

HTML files

See 'Files' section for details

db.py

Handles the database, including login information and saving attendance behavior

__init__.py

Flask app, handles front-end navigation and operations

database.db

Contains user information including attendance

FILES

base.html

header

create.html

Asks for profile information for first time setup:

Username, password, type of account, admin password (if teacher account)

Has a link to the login page

login.html

Form for username and password

Has a link to the create account page

profile.html (student / section leaders)

Includes a form at the top to register a class:

- Class, class code, and section

Contains a calendar that presents attendance record and important events

Contains overall attendance and grades

attendance.html (section leaders / teachers)

Contains master list of students in each class period for each day

Section leaders / teachers can mark attendance

Teachers can also remove students from class

excuse.html (section leaders / teachers)

Section leaders can excuse student

class.html (teachers)

A form for teachers to create new classes

- course code/section, class password, and band/chorus

student.html (teachers)

Sees any specific student profile

Can change their grade and change their attendance

home.html (teachers)

Allows teachers to add section leaders

profile.js

For profile.html and student.html

Hover over each date to see excused/unexcused and reason

main.py

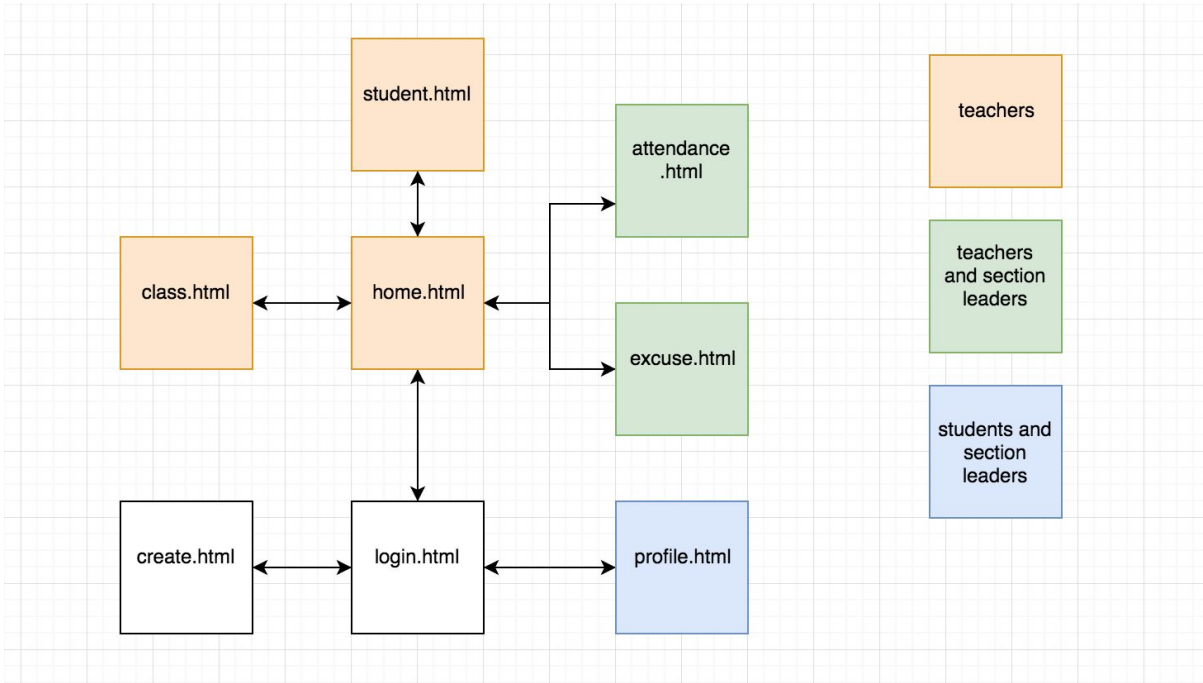
- **add_session(username, password)** - creates a session cookie for the current user with valid credentials or returns error message
- **is_null(username, password, confpw)** - checks to see if any of the username or password submissions were null
- **logout()** - Deletes a session key of a user, if one is logged in
- **login()** - provides functionality for the /login route
- **create()** - provides functionality for the /create route
- **profile()** - provides functionality for the /profile route
- **home()** - provides functionality for the /home route
- **attendance()** - provides functionality for the /attendance route
- **excuse()** - provides functionality for the /excuse route
- **class()** - provides functionality for the /class route
- **student()** - provides functionality for the /student route

db.py

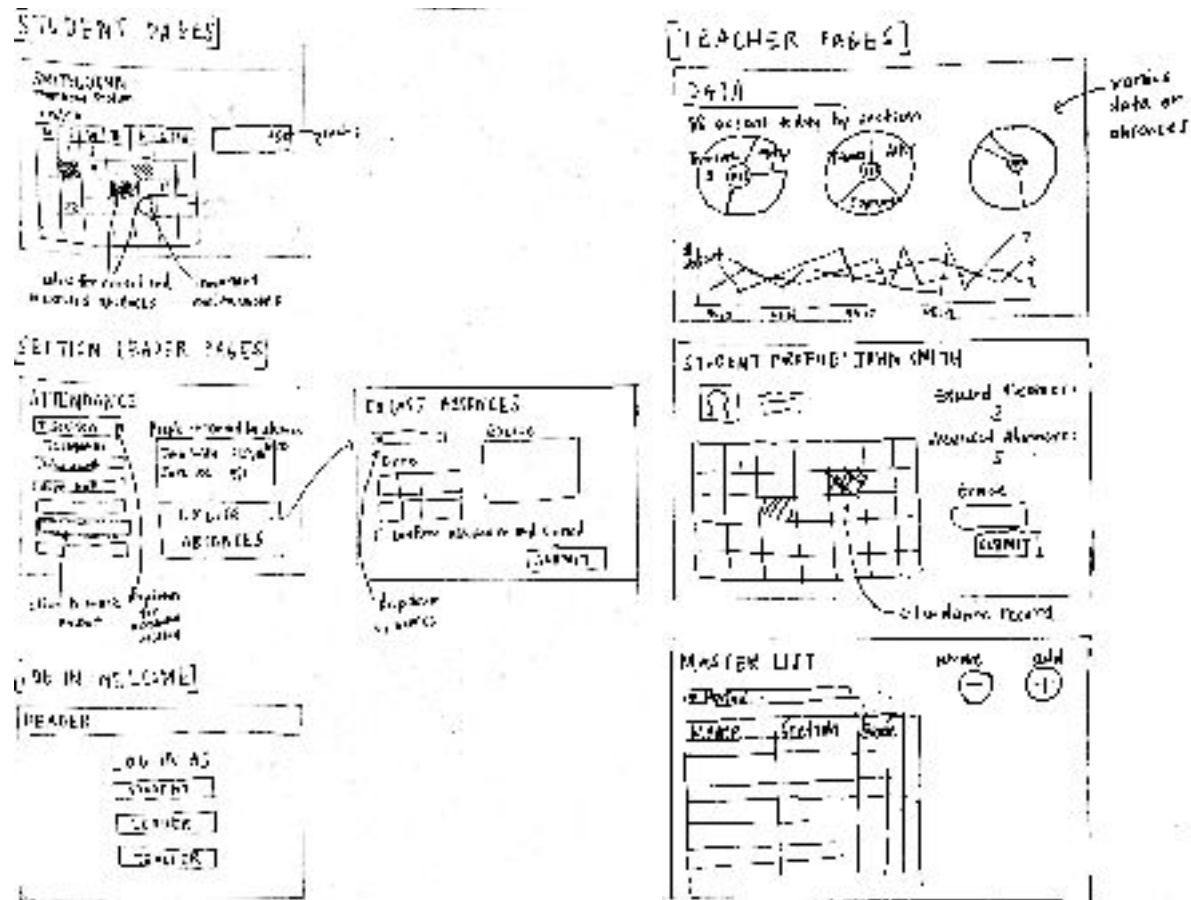
- **login(username, password)** - Logs user in if username and password are correct
- **encrypt_password(password)** - Returns SHA-256 version of password
- **create_account(username, password)** - Creates a new entry in .profiles table
- **does_username_exist(username)** - Returns true if username exists, false otherwise
- **get_leaders(coursecode)** - Returns a list of a class's section leaders
- **get_accounttype(username)** - Returns the account type for a specific email

- **add_attendance(username, date, type, reason)** - Appends entry into attendance table if email and date doesn't exist in table, otherwise updates it
- **get_classtype(coursecode)** - Returns class type for a specific class
- **authorize_class(coursecode, password)** - Adds the student into class if correct information given
- **create_class(teacher, coursecode, password, type)** - Adds a class into the class table
- **add_leader(coursecode, username)** - Gives a user leader access
- **remove_leader(coursecode, username)** - Removes a user from leader access
- **add_student(coursecode, username)** - Adds student to a class
- **remove_student(coursecode, username)** - Removes student from a class
- **get_grade(coursecode, username)** - gets student's grade in class
- **change_grade(coursecode, username, grade)** - changes student's grade in class
- **student_present(username, date, course)** - returns if student was present
- **delete_attendance(username, date, course)** - removes absence from attendance table

SITEMAP



PAGE LAYOUTS



DATABASE SCHEMA

Profile Table

The profile table contains 4 fields: email, full name, password, and account type (S for student, L for student leader, T for teacher)

Email (Primary Key)	Password *encrypted	Full Name	Account Type
wheeler@stuy.edu	adfcaegasd	Raymond Wheeler	T
kli16@stuy.edu	shrek	Kevin Li	L
gvidali@stuy.edu	1234goal	Giorgio Vidali	S

Attendance Table

The attendance table contains 4 fields: student email, date, and attendance, and reason. The only attendance information that will be stored is absence (E for excused and U for unexcused). The reason column will be blank if unexcused.

Student Email	Day	Course	Attendance	Reason
kli16@stuy.edu	2017-10-03	UVS11-01	E	Debate tournament
gvidali@stuy.edu	2018-05-18	UVS11-01	U	

Classes Table

The classes table contains 5 fields: teacher email, course code, course password, class type (B for band or C for chorus), and a list of the section leaders.

Teacher Email	Course Code (Primary Key)	Class Password *encrypted	Type
wheeler@stuy.edu	MKS66-01	asdfghjkl	B
shamozov@stuy.edu	UVS11-01	abc123	C

Leaders Table

The leaders table contains 2 fields: course code and section leader emails. There are multiple slots per course for multiple section leaders.

Course Code	Section Leader
MKS66-01	jxu10@stuy.edu
MKS66-01	kli16@stuy.edu
UVS11-01	gvidiali@stuy.edu

Enrollment Table

The enrollment table contains 4 fields: course code, student email, name, and grade. There are multiple slots per course for multiple section leaders.

Course Code	Student	Name	Grade (int)
MKS66-01	jxu10@stuy.edu	Jacqueline Xu	100
MKS66-01	kli16@stuy.edu	Kevin Li	98
UVS11-01	gvidiali@stuy.edu	Giorgio Vidali	65

TENTATIVE TIMELINE

Friday 5/11: Basic setup of repo

Tuesday 5/15: Finish design document

Friday 5/18: Finish login page and begin student profile

Tuesday 5/22: Finish student profile and class pages

Monday 5/28: Finish section leader page and master list

Friday 6/1: Finish Teacher page

Friday 6/6: Add CSS and design to make it more user friendly

TASK ASSIGNMENTS

Giorgio Vidali: Project Manager

Cynthia: CSS and Calendar API

Kevin: D3 data visualization

Jackie: front end HTML and and Database