FINAL PROJECT
Lens.

Pd 9 | Projection of Despair in Four Stages
James Smith, Max Korsun, Angel Ng, Connie Lei

WHAT DOES IT DO?

Our project is an active lens diagram generator. You will be able to adjust the object distance, object height, concave vs convex lens, and focal length and watch the light vectors and image vector respond to these changes. Numbers will be displayed as well. Will zoom in and zoom out based on image height and distance relative to lens. Users who are logged in will have their sessions actively saved with each change. The session is sharable for users who are logged in through the link. These diagrams will also be able to be downloaded as images.

Stretches - Viewing other people's profiles, adjusting the concavity of the lens visualization to match the actual focus, drawing picture for object and having the image match it.

COMPONENTS

HTML files

See 'Files' section for details

Stylesheets

Bootstrap's as well as our own

Python files

See 'Files' section for details

SQLite Database

See 'Database Schema' for details

FILES

template.html

login/logout button, profile button, lens button, and homepage button on a toolbar, and footer with team info

home.html

Explains how to use the program and is the hub for viewing projects

profile.html

Displays owned diagrams, username, and option to change password

login.html

Form for login and creation of accounts

lens.html

For viewing images and lens'

style.css

To make the website look good

main.js

Provides basic ajax calls and needed functionality

bootstrap.min.css/bootstrap.min.js

Incorporating Bootstrap

Main.py - Flask app, handles front-end navigation and operations

- **add_session(username, password)** - creates a session cookie for the current user with valid credentials or returns error message
- **logout()** - Deletes a session key of a user if one is logged in
- **root()** - provides functionality for the home.html for the root route
- **login()** - provides functionality for the /login route
- **profile()** - provides functionality for the /profile route
- **lens()**-provides functionality for the /lens route

Db.py - Handles the database, including adding lens saves as well as user account information

login(username, password) - Logs user in if email and password are correct

- **encrypt_password(password)** - Returns SHA-256 version of password
- **change_password(old_password, new_password)** - changes password to new_password if old_password matches the account password
- **create_account(username, password)** - Creates a new entry in .profiles table
- **create_session(account_name, o_dist, o_height, focus)** - Creates a new session with a unique hashcode identifier
- **get_session**(**hashcode**) - returns all information in a dictionary about the session
- **get_owned_sessions(username)** - returns hashcode identifiers of all owned sessions
- **update_session(hashcode, o_dist, o_height, focus)** - updates data in session

SITEMAP

DATABASE SCHEMA

Users

id	username	password*
0	user	pass123
1	user3	awd
2	user1	yeet

*SHA-256 Encrypted

Sessions

hash_id	id	o_dist	o_height	focus
123456789	4	32m	5m	3
123456788	2	900m	300m	-90

TASK ASSIGNMENTS

James: Front-end Javascript
Max: Project Manager, apache, and wildcard
Angel: HTML, CSS, and wildcard
Connie: Database and Flask

TIMELINE:

- Setting up Apache - May 19th
- Database Functionality - May 24th
- HTML Templates - May 24th
- Lens MVP Diagram - May 24th
- Downloadable diagram - May 25th
- Profile functionality - May 26th
- Saving sessions - May 28th
- Styling - May 28th
- Complete lens Diagram - May 28th
- Finished - June 6th