Chime In

Please follow these instructions to run the project.

Setup and requirements

Python Environment

Our project relies on different python libraries in order for it to function. To make setup easier, we have included an `environment.yaml` file. This will allow you to create your own miniconda virtual environment. We recommend you look into [their installation instructions](#).

You can create and activate this environment with the following:

```
conda env create -f environment.yaml
conda activate lambda-env
```

Or you can create your own environment and install the packages with your preferred python venv manager by installing the following libraries:

```
numpy >= 1.26.0
pillow >= 10.0.1
pytesseract >= 0.3.10
matplotlib >= 3.7.2
pymongo >= 4.5.0
requests >= 2.31.0
fastapi >= 0.103.2
pydantic >= 2.4.2
uvicorn-standard >= 0.23.2
opencv-python-headless >= 4.8.1.78
```

Database Variables

To then connect to the database you will need to set the environment variables for the username and password as

```
export DB_USER=<database-username>
export DB_PASS=<database-password>
```

> [!NOTE] If you need access to the server that we are using please message Lucas Hicks

Node Requirements

We recommend you install Node Version Manager(NVM) to maintain a correct node version. Once NVM is installed we can then run

```
nvm install latest
nvm use latest
```

> [!NOTE] If you don't wish to use NVM then use your chosen package manager to install Node Latest.

Then we can install the required node packages with:

```
npm install react --save
npm install -g serve
```

This will install all of our node requirements so that you can run the display app.

Build Instructions

Generally for running this program, you should first set up a camera so that it is viewing the whiteboard calendar. Ensure the image taken and the calendar itself are as level as possible. Then run the calibration script. Next you will need multiple terminals as we will run many server based and constant up apps. These will run as (in the following order):

1. The database
2. The display
3. The microcontroller communications (After connecting the microcontroller)
4. The input capture

Then everything will be running and communicating with each other.

Developers

Developed by Alex Viller, Lucas Hicks, David Jeong, Dylan Fleming, Maya Baxter, and Ali Laherty at the University of Queensland. Thanks to Rostislav Gusev and the rest of University of Queenslands DECO3801 teaching team for all the support.

> [!IMPORTANT] If any further details are required, please contact a member of Team \Lambda