Instructions For a Working Concerto Build:

- How to install rbenv and Ruby 2.6:
    - https://github.com/rbenv/rbenv?tab=readme-ov-file#basic-git-checkout
        - *git clone https://github.com/rbenv/rbenv.git ~/.rbenv*
        - *~/.rbenv/bin/rbenv init*
        - *rbenv install 2.6.0*
        - Run *ruby -v* to make sure that the right version was installed.
        - One of these:
            - Set default Ruby version for this machine:
                - *rbenv global 2.6.0*
            - Set default Ruby version for this directory:
                - *rbenv global 2.6.0*
        - *gem install bundler*

- How to run the Docker Image (DOES NOT WORK):
    - Below are the modified instructions I used to get the Docker Concerto build in a somewhat working state. Unfortunately, I ran into a Database connection issue that I could not fix. After trying for a couple of weeks, Jonathan and I decided that we should focus on the Raspberry Pi installation instead. Additionally, the Concerto wiki even states that the Docker image is a work in progress.
    - NOTE: Make sure that all of your files have the correct permissions. When trying to execute *bundle exec rake db:migrate,* I kept receiving an error that stated I was using the wrong version of Ruby. It also recommended that I run *bundle gem install*. To fix this, I just added R/W permission to all of the files within my Concerto project by using the CHMOD command.

- Dependences:
    - Ruby 2.6
    - Rubygems
    - Imagemagick, GhostScript, Poppler-Utils
    - LibreOffice
    - Webserver (Apache/Unicorn/Thin/Nginx)
    - Rack interface to the webserver (Passenger, FastCGI)
    - ActiveRecord-compatible database (Mysql, SQLite, Postgres)
    - Nodejs as the javascript engine (as of version 2.4.0)
- Download Docker Desktop:
    - https://www.docker.com/products/docker-desktop/
    - If needed, here's the link to the Concerto base image:
        - https://hub.docker.com/r/concerto/concerto-base

- Now follow the instructions listed in the README:
  - https://github.com/concerto/concerto
    - *git clone http://github.com/concerto/concerto*
    - Edit your bash file to include:
      - *export PATH="$HOME/.rbenv/bin:$PATH"*

        eval "$(rbenv init -)"
    - *cd concerto*
      - Comment out the following line from docker-compose.yml since there’s a authentication bug.
        - *command: --authentication-policy=mysql_native_password --character-set-server=utf8mb4 --collation-server=utf8mb4_general_ci*
    - *bundle install --path vendor/bundle*
    - *bundle exec rake db:migrate*
    - *bundle exec rake db:seed*
    - *docker build -t concerto .*
    - *docker-compose up*

- How to create a local/LAN instance of Concerto (WORKS):
    - NOTE: This is the Concerto installation that we eventually ported to a Raspberry Pi since the Docker build could not be fixed.
    - https://github.com/concerto/concerto/wiki/Installing-Concerto-2
        - *sudo apt-get install -y build-essential apt-transport-https libapache2-mod-passenger ruby-full ruby-dev libruby imagemagick ruby-rmagick libmagickcore-dev libmagickwand-dev libssl-dev zlib1g-dev libsqlite3-dev default-mysql-server libpq-dev default-mysql-client ruby-mysql2 default-libmysqlclient-dev apache2 libxslt1-dev nodejs git*
        - *gem install bundler*
        - *sudo add-apt-repository ppa:libreoffice/ppa*
        - *sudo apt-get update*
        - *sudo apt install -y libreoffice ghostscript libgs-dev gsfonts poppler-utils*
        - *git clone https://github.com/concerto/concerto.git*
        - *cd concerto*
        - *bundle install --path vendor/bundle*
        - *bundle exec rake db:migrate*
        - *bundle exec rake db:seed*

- LocalHost:
    - *RAILS_ENV=production bundle exec rake assets:precompile*
        - If you wish to create a development build, just replace *production* with *development*. The inverse is true for the LAN example below.
- LAN:
    - *RAILS_ENV=development bundle exec rails server -b* <your IP Address>
        - NOTE: Don’t include “<” or “>” in the command

What I’ve done:

- NOTE: I did not make any code changes, so everything listed on this document reflects the steps I’ve taken to create a working Concerto build and make progress on the Docker image. Also, please keep in mind that we didn’t even have a working Concerto build, nor knew how to create one, at the beginning of this semester. The main goal of this project has always been to restore the defunct Concerto displays in Lally and Amos Eaton.

- Developed a script that launches Concerto when the Raspberry Pi turns on.

- Made progress on the Docker build (originally, we couldn’t even get past the *bundle exec rake db:migrate* step.)

- Created a Concerto build for the Raspberry Pi.

- Debugged issues Jonathan and I ran into while hosting Concerto on the Raspberry Pi (installing Ruby Gems, CSS formatting issues, adding missing dependencies, upgrading depreciated tools/dependencies Concerto needs, etc.)

- Helped Jonathan set up SSH so that we could work on the Raspberry Pi remotely.

- **Installed the Raspberry Pi, which is hosting a working Concerto build, in Lally Hall (stop by and check out our work!)**

Roadmap for the future:

I plan on returning to work on the Concerto project next semester, so I've already discussed with Jonathan some of the future goals we want to accomplish

- Monitor the Concerto installation in Lally Hall.

- After/if the ethernet port in Amos Eaton Hall (AE) is switched on, Jonathan and I will implement another Concerto in AE.

- Read through the Kerberos documentation, so that we can implement a way to authenticate Concerto users securely.
    - https://www.cs.rpi.edu/academics/courses/fall04/os/kerberos.html

- Figure out why our Concerto installation is being blocked by Google/YouTube. For some reason, the Raspberry Pi hosted version cannot display YouTube videos, but local builds (on our laptops) can.

- Write a manual detailing how to create screens (what Concerto displays), so that faculty members and our peers can utilize Concerto.

- For some reason, Concerto does not send an email when a user asked for a password change request, so this is a major bug that we’re currently working on.