**Picard Documentation**

**Building The Picard in Docker Container:**

After pulling down the code from the github cd into the directory, download docker and run the following commands:

docker pull nginx:latest

docker build -f Dockerfile.prod -t docker-image-prod .

docker run --env-file ./.env -d -p 8080:80 --name container-name image-name

(Port for NGINX on local must allocate 3001:3001 for the container to run)

docker-compose up --detach to run it so we don't have to use ctrl + c to stop it. It keeps it running

**Work Done:**

Our group worked on building the GUI for the Picard after the first group setup the server and hardware. The GUI includes:

1. Sign up Page
2. Sign in Page
3. Home Page
4. Account Page
5. Create Experiment Page
6. Algorithm Description Page
7. Add Algorithm Page
8. Algorithm Parameters Page
9. Results Page
10. **Sign up page:**

The sign up page works fully by asking the user to use an email address and a strong password with a description of what should be included in the password.

1. **Sign in page:**

The sign in page works fully by asking the user who signed up to enter the email address used and the password created so the user can be able to login to the website.

1. **Home page:**

When the user logs in successfully, they will be able to see the home button, which will always redirect the user back to the home page; the account button, which will redirect the user to the account page; the create experiment button, which will redirect them to the create experiment page; and last but not least, the algorithm description button. Which will redirect the user to the algorithm description page, and finally the results button, which will redirect the user to the results page.

1. **Account page:**

The account page works fully, showing the user the email address signed in and there's a logout button that the user can click to logout.

1. **Create Experiment page:**

From the home page the user will be able to access the Experiment page by using the Create Experiment button. Once on this page the user will be able to set up an experiment. First input a number for the amount of driver memory you would like to use, followed by the executor memory as well. Then select the number of executor cores, the number of nodes, and the memory overhead. All above input fields are integers. Finally, you will select the algorithm you wish to use for the experiment from the dropdown. To the left of this you will see the run experiment button which will redirect the user to the algorithm parameters page. At the bottom you will see a large text box with the selected environment parameters. To the right of this you will see the save and delete buttons. The save button allows the user to save the currently selected environment parameters. The delete button will clear all of the environment parameters from the input fields and the text box at the bottom.

1. **Algorithm Description page:**

The algorithm description page covers the details of what each algorithm’s purpose is. Once an algorithm has been added you will notice it will add the algorithm to the page as well.

1. **Add Algorithm Page:**

The add Algorithm page is accessible through the Algorithm description page by clicking the upload algorithm button. After getting redirected, the user will be able to add an algorithm name and maximum of 30 parameters to the backend after pressing the add algorithm button.

1. **Algorithm Parameters Page:**

After clicking run experiment on the create experiment page the user will be redirected to the algorithm parameters page. On this page the user will fill out a number of input fields that are further parameters in the experiment. The number of input fields will change relative to the algorithm of the users choosing. Once all desired input fields are filled, the user will hit the submit algorithm button at the bottom of the page and a text file will be generated that will be downloaded locally.

1. **Results page:**

From the home page the user is able to access the Results page using the Past Results button. Once on this page the user will be able view a number of past experiments and their results. The user will also be able to download any results of their choosing. You will also see two buttons at the top-left of the page, the home button which redirects back to the home page and the account button which redirects to the account page.

**Future work and bugs:**

Certain aspects of the application are not fully complete. There is a scroll bar issue on certain pages, such as the algorithm description. Some other minor rendering issues occur, such as the upload algorithm button. The one-button upload file is just a placeholder, and certain aspects of the functionality are missing. The specific rendering issues are minor bugs but will need to be fixed prior to deployment. There is currently no duo authentication; however, we have implemented Google Authentication in the past and removed it for demonstration purposes. A form of duo authentication should be implemented prior to deployment and is fairly simple with the use of Firebase, as we discovered previously. What’s more, the submit algorithm works, but if the inputs go extra high, it saves the files with 0 bytes. Another aspect is the results page is a placeholder as well. Ideally, you can have those results pulled down from the cloud, and display them for each user.