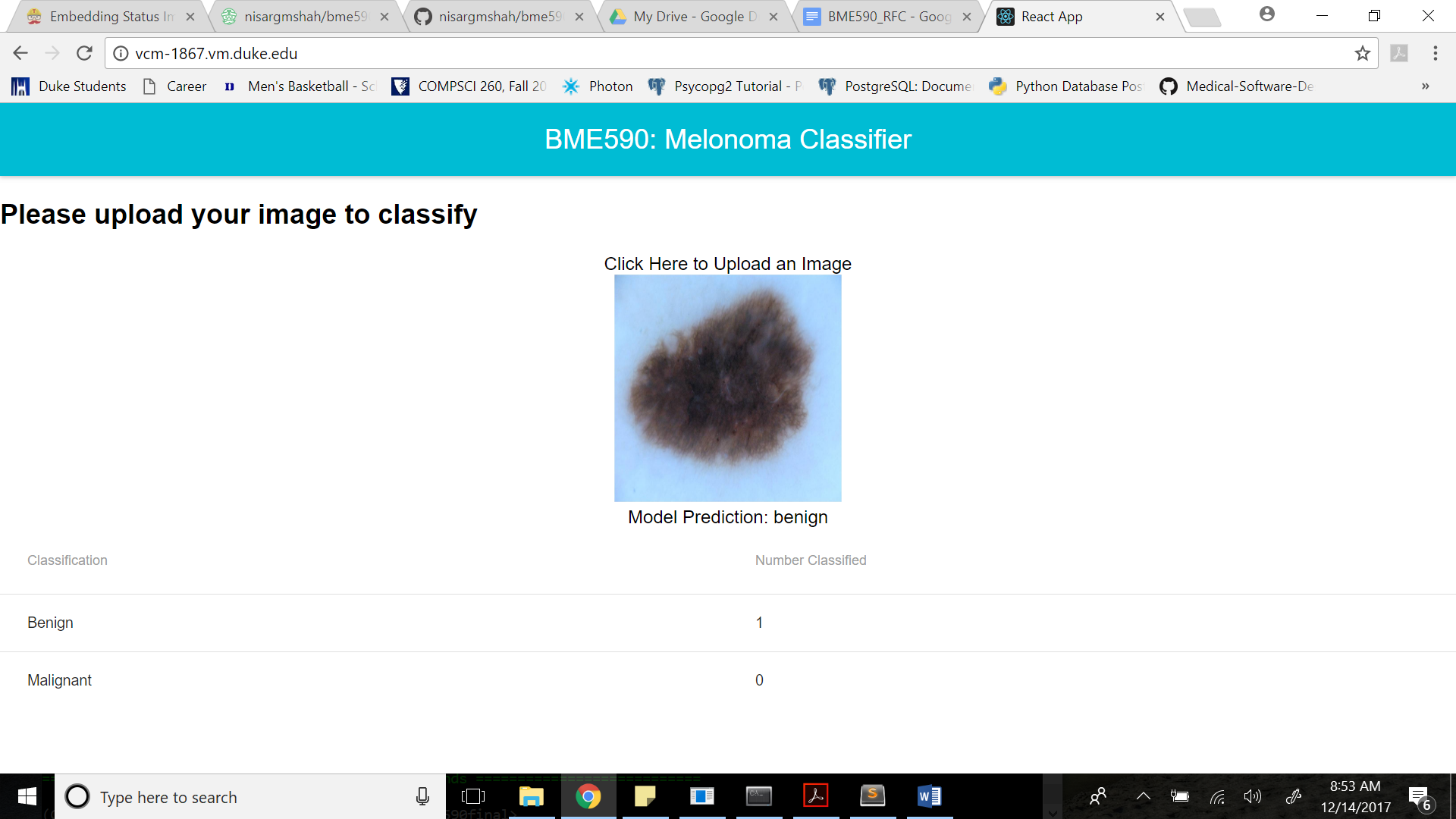
**Title:** Skin Lesion Classification using Neural Network Model

**Authors:** Nisarg Shah, Ryan Spears, Garren Angacian, Joshua Khani

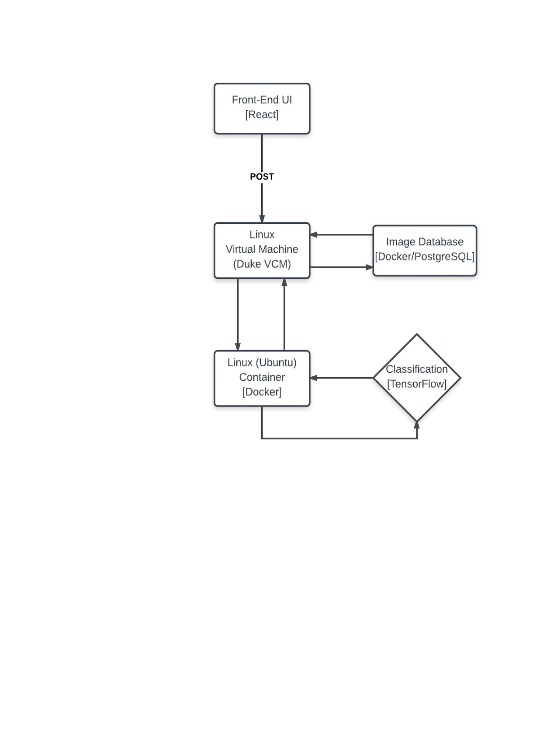
**Objective:** The goal of this project is to develop a web service which accepts an image of a skin lesion and returns a prediction as to whether the lesion is malignant or benign.

**Approach:** Our approach entails a frontend web service which can be accessed through the browser. Users interested in classifying a skin lesion can upload an image of the lesion through our web service. The image is then processed and posted to a server running a Docker container that contains a TensorFlow image and can call methods from the TensorFlow library. A neural network model trained on thousands of labeled skin lesion images can then be used to determine the likelihood that the image contains a malignant or benign lesion. The server then stores the results of the classification in a SQL database and returns a prediction to the user.

*Frontend Web Service:* The frontend web service was built using React, a Javascript library.



*Backend Server:*



*Image database (SQL):*

**melonoma\_images**

|  |  |  |
| --- | --- | --- |
| Attribute | Data Type | Description |
| Image\_ID | ID (Primary Key) | Unique identifier for each image |
| Model\_Prediction | String | The classification that the model gives an image: benign or malignant |

**Results:**

The web application can be used to upload images of skin lesions and determine if they are benign or malignant for melanoma. However, currently the images are delayed by 1, so the second image posted gives the result of the first image posted and so forth. This is due to a React issue that was not able to be resolved. The database is able to store data on the number of benign and malignant images, however the connection to the database is not working when the docker container is running on a screen. This is a known Docker and Postgres issue according to Owen which could be due to the database not being fully running by the time it is called.

**Future directions:**

More unit tests and database functionality will be created in the future in addition to a more accurate model. Statistical tests will also be done once the model’s accuracy is improved. The front end will also have more features for interactivity with the user and the ability to get and visualize more data.