In this first project you will explore the process of creating your own neural network model, training it, and evaluating it. You are going to train a deep multilayer perceptron on the CIFAR-10 dataset. Read more about this dataset here, including how to load it into your workspace. The dataset will come with training and testing data. You must use the training data to create a validation dataset, like we did in class. Use this set for validation during the fitting process. The test set should only be used for final evaluation after training.

You are only allowed to use the following types of layers: `Input`, `Flatten`, and `Dense`. These are the layers introduced in Chapter 10 for the classification examples. You are not in a competition to create the best possible CIFAR-10 classifier, so there's no need to scour the internet looking for the best image classifier models. The primary goal of this project is to understand how to build and train neural network models on your own.

Carefully review the textbook's recommendations for tuning hyperparameters. Then, through repetitive manual tuning of these hyperparameters, find the highest classification accuracy your manual tuning is able to determine. Do not spend an absurdly long time doing this part of the project. I recommend that you try to create a clever function that will handle this manual hyperparater tuning process for you. That way you can run your code on one (or more) of the Data Science Lab's computers, while you leave and take care of other business.

After you are satisfied with your manual tuning, try tuning the hyperparameters with Keras Tuner. Be sure to use the `ModelCheckpoint` and `EarlyStopping` callbacks to make the fitting process as efficient as possible. If you are able to get TensorBoard running on either your own computer or DS Lab computer, I recommend using it to plot learning curves.

Upon completion, you must write a report that completely details your project, including your model architecture, your manual training results, Keras Tuner results, and so on. I expect your report to have multiple figures that capture your training metrics. These figures should be discussed in your report. You do not need to include your code as an appendix to the report. However, your report should have an appendix that summarizes the textbook's recommendations and rules of thumb for tuning hyperparameters. This appendix need not be lengthy, but it should be thorough and useful to someone who is not familiar with hyperparameter tuning.

Email your code and report to me no later than 11:59 PM, February 9, 2026. I encourage you to develop your project using Jupyter Lab, but it is not required. You may name your files whatever you feel is appropriate.