**Step 4: Write a Report on the Neural Network Model**

For this part of the assignment, you’ll write a report on the performance of the deep learning model you created for Alphabet Soup.

The report should contain the following:

1. **Overview** of the analysis: Explain the purpose of this analysis.

* The purpose of this analysis is to build and evaluate a deep learning model to predict whether applicants for funding from Alphabet Soup will be successful. The analysis involves preprocessing the data, defining and training a neural network, and optimizing its performance to achieve an accuracy target of at least 75%.

1. **Results**: Using bulleted lists and images to support your answers, address the following questions:

* Data Preprocessing
  + What variable(s) are the target(s) for your model?
    - The target variable for the model is IS\_SUCCESSFUL, which indicates whether the funding application was successful (1) or not (0).
  + What variable(s) are the features for your model?
    - APPLICATION\_TYPE
    - AFFILIATION
    - CLASSIFICATION
    - USE\_CASE
    - ORGANIZATION
    - STATUS
    - INCOME\_AMT
    - SPECIAL\_CONSIDERATIONS
    - ASK\_AMT
  + What variable(s) should be removed from the input data because they are neither targets nor features?
    - EIN (Employer Identification Number) – a unique identifier that doesn't contribute to prediction.
    - NAME – assuming it's a non-informative identifier.
* Compiling, Training, and Evaluating the Model
  + How many neurons, layers, and activation functions did you select for your neural network model, and why?
    - First Hidden Layer: 7 neurons with ReLU activation – to capture initial patterns.
    - Second Hidden Layer: 14 neurons with ReLU activation – to capture more complex interactions.
    - Third Hidden Layer: 21 neurons with ReLU activation – to capture deeper patterns.
  + Were you able to achieve the target model performance?
    - No, despite implementing optimizations like Batch Normalization, Dropout, and Learning Rate Scheduler, the highest validation accuracy achieved was 72%, which is below the target of 75%.
  + What steps did you take in your attempts to increase model performance?
    - Implemented Dropout Layers: To reduce overfitting by randomly dropping neurons during training.
    - Added Batch Normalization: To stabilize and speed up the training process by normalizing the inputs of each layer.
    - Used Learning Rate Scheduler: To adjust the learning rate dynamically, improving convergence.
    - Hyperparameter Tuning: Experimented with different numbers of neurons, layers, and activation functions to find the best architecture.
  + Early Stopping: Monitored validation loss to stop training when performance stopped improving, preventing overfitting.