In this document I have penned down my approach towards this problem (4.2 NLP):

- **Hrithik Nambiar**

Task 1,2:

The first step towards this task, was analysing the data and cleaning it. In the dataset there were 238 nan values under 'Posts' and these had to be removed. After this, I decided to merge both the title and posts together as they described the same thing and decided to feed this as the input, labelled with the flairs. After processing the data, I stored it into train.csv and val.csv respectively. For data processing in English, I used Spacy. SpaCy is an open-source software library for advanced natural language processing. Then Torch text was used for further processing, for e.g.: converting to lower case etc.

For embedding I have used GloVe embeddings, which is a very powerful word vector learning technique. After these steps we are ready to build an RNN based model. The broad idea is that we input the concatenation of title and body of the post and try to predict flair using a softmax classifier.

Next a LSTM encoder is used. An LSTM Autoencoder is an implementation of an autoencoder for sequence data using an Encoder-Decoder LSTM architecture.

As mentioned in Task 2, I then added an Attention layer to the LTSM outputs. The final model uploaded, is the one with this layer. The baseline accuracy was lower than by this model. Next comes, the softmax classifier which classifies it into the 15 classes (Note : logits.view(-1,15) in the code).

The optimizer used in this model is the Adam optimizer, and the hyperparameter's (hidden size, number of layers, dropout etc)were tuned to fit the maximum accuracy after multiple attempts.

The final accuracy on the validation data is 57.33%.

Task 3:

Note that this has not been completed.