**Django service:**

1. Follow readme and get the app running
2. All the models are present in the models folder
3. Distron.py is deprecated
4. The doctor nodes is processed using metamap invoked as a http request in mmlrestclient.py
5. Model\_config.py consists of the neural network
6. views.py is deprecated
7. EpicInterface.py calls mml and processes the required doctor notes
8. Deploy it along with the web-application deployment
9. Run the application on **0.0.0.0:8000** and see it running on the the aws\_site:8000
10. As for node service, 8000 has to added in the inbound port rules
11. Pm2 is only working locally for Django, so need to check for an alternative to run this application background as the application is deployed on the machine.
12. **This is important** - python3 manage runserver 0.0.0.0:8000

**Next steps:**

1. Connect the EHR APIs to the Django service for every EHR system - Tejvir In Progress
2. AI model input data feeding functionality - needs attention
   1. AI model library needs data to come in a certain way
   2. See Django service EpicView
3. Integration with new models, when AI team is finished
4. Find an alternative to PM2. Something that can automate the Django service
5. Look into Django API security
6. Develop API for the future models for other systems(UI/node backend) to fetch json output predictions
7. Logging