# Practicum Sprint #3

# Mental/Physical Illness Chatbot

Tusheet Goli, Tejas Pradeep, Akshay Sathiya, Pranav Khorana, Sanket Manjesh, Rahul Chawla
tgoli3@gatech.edu, tpradeep8@gatech.edu, asathiya6@gatech.edu, pkhorana3@gatech.edu, smanjesh3@gatech.edu, rchawla36@gatech.edu

## 1 ACCOMPLISHMENTS THIS WEEK

### 1.1 Akshay Sathiya's Progress

This week, Akshay completed the first task listed in the Practicum Sprint #2 submission (preprocess data from datasets) for the physical diseases dataset (Patil, 2020). This textual dataset must be processed into numerical form so ML models can train on it. Akshay wrote the code to convert each row of the original dataset into a numerical feature vector and label. Each element of the feature vector corresponds to a unique symptom in the dataset and is set to 1 if the symptom is present, 0 if not present. The label is an integer corresponding to a specific disease in the dataset. The processed dataset (feature vectors and labels), the mappings of disease names to labels, and the list of unique symptoms present in the dataset are saved in a CSV and JSON file for future reference/use.

Akshay has pushed his Python code, the CSV and JSON output files, the physical diseases dataset (Patil, 2020), the Twitter dataset (Merin S, 2020), the emotions dataset (Praveen, 2020), and other project files (README.md, .gitignore, virtual environment requirements text file) to the GitHub repository. They can all be seen on the *physical_illness_data_analysis* branch.

### 1.2 Pranav Khorana's Progress

This week, Pranav got started on the 4th task listed in the Practicum Sprint #2 submission (designing a mobile application) by setting up the React Native app that will serve as the UI for our chatbot. Pranav chose React Native because it enables us to easily deploy the app on both Android and iOS devices. In order to accomplish this, he used Expo, a framework and platform that has tools that help users develop, build, and deploy apps ("Introduction to Expo"), to initialize the

React Native App with Typescript configuration. Pranav also downloaded an Android Emulator on his computer so that testing the mobile application would be quicker and easier to debug. His Android Emulator is Pixel 3 XL that uses the Android 8.0 operating system.

### 1.3 Rahul Chawla's Progress

This week, Rahul worked on task #4, as listed in Practicum Sprint #2. Rahul set up a PostgreSQL database, and deployed it to Heroku. He chose PostgreSQL due to its robust feature set, such as its object-oriented database features, as well as its SQL standards compliance, which all the group members are already familiar with. Rahul chose to deploy the database on Heroku due to its ability to handle all infrastructure problems and allow the group to focus solely on the code. In its current state, the server allows for 20 simultaneous connections, 10,000 rows for tables, and 8 megabytes of data, which should be more than sufficient for the scope of this project.

### 1.4 Tusheet Goli's Progress

This week, Tusheet started working on setting up the core backend and helped in developing the API structure that is going to be used by the app. He helped design the deployment structure for the application, i.e., incorporated a docker containerized environment for individual services. These containerized deployment environments enable the application to be scaled based on these individual containers. We plan on using Heroku for deploying this application as Heroku provides a natural way to integrate our application with a PostgresSQL database hosted by Heroku. In addition to this, he worked closely with the API layer to design the basic structure of the API that will be used to communicate with these individual services. Tusheet mainly concentrated on setting up the basic software components of the application and other service that we plan on using in this sprint. Thus, Tusheet helped in developing a robust cloud-hosted backend with individual scalable docker containerized environments deployed on Heroku.

### 1.5 Tejas Pradeep's Progress

This week Tejas setup the python Flask development environment as well as setup the basic API structure. Tejas got started on task #1 listed in the Practicum Sprint #2 submission (designing the backend) by creating a backend in Python

Flask containing REST API methods without implementation. Implementation of these methods shall happen at a later sprint once the database is set up and the analysis models are set up. He designed the API to contain two models to facilitate the app's behaviors - User API (contains logic for user information), Chat API (Contains logic to interact with various systems to facilitate conversation). The work done here is pushed to the backend_design branch on the github repository.

### 1.6 Sanket Manjesh's Progress

Sanket was able to set up the server to connect both the database and the front-end application as mentioned in Sprint #2. Sanket decided to use Node.js as the library to set up the server. Sanket chose Node.js because it was relatively easy to use compared to other frameworks and very quick to get started with and send requests/responses. Sanket's next steps are to ensure that the Node.js server is connected to both the front-end application and the database and that requests and responses can smoothly be sent between each.

## 2 CHALLENGES ENCOUNTERED

### 2.1 Sanket Manjesh's Challenges

There could be future challenges dealing with the scalability of setting up the server with Node.js in the near future, especially if the group ends up using a large amount of data and many requests/responses need to be sent. If these scalability issues arise, the group might consider using another framework, such as Java SpringBoot, to set up the server. We might also switch to a more complex microservice architecture if needed to break down components of our server and make it more efficient.

### 2.2 Tejas Pradeep's Challenges

Since the backend is a core aspect of the app that interacts with all other parts of the app it was a little difficult to design it without much clarity on the other aspects of the app like the database design and the ML models. Currently he made decisions based on intuition and plans discussed for the other aspects, this backend design may have to be adapted once those sections of the code are completed.

## 3 FUTURE PLANS

The main aium of this spring was to set up the foundations software architecture for the application and any other services we planned on using. In addition to this, we worked on some preliminary data process and cleaning for a couple of our datasets. We have made good progress this sprint as per the plans on our Project Proposal document. We are hence going as per schedule and will continue working on our tasks based on the proposed schedule.

## 4 REFERENCES


1. Merin S, S. (2020, April 17). *Twitter Emotion Analysis*. Kaggle. Retrieved March 6, 2022, from https://www.kaggle.com/shainy/twitter-emotion-analysis/data
2. Patil, P. (2020, May 24). *Disease Symptom Prediction*. Kaggle. Retrieved March 6, 2022, from https://www.kaggle.com/itachi9604/disease-symptom-description-dataset
3. Praveen. (2020, April 16). *Emotions dataset for NLP*. Kaggle. Retrieved March 6, 2022, from https://www.kaggle.com/praveengovi/emotions-dataset-for-nlp
4. *Introduction to expo*. Expo Documentation. (n.d.). Retrieved March 13, 2022, from https://docs.expo.dev/