Background:
Biophysical models of the brain often simulate thousands of neurons spiking in a network; however, these models often are complex and have long run times. The simplest form of modeling the neuron is the integrate-and-fire model. In this representation, the neuron's membrane potential is modulated by presynaptic inputs, then sends a signal to it's postsynaptic outputs when a membrane threshold is reached. Although biophysical relevance of the neuron itself is diminished with this approach, biophysical properties of spiking networks arise.

In this package, a model of spiking neural activity over 5 layers of the cortex will be generated. Number of neurons in each layer as well as the connectivity between each layer and other layers can be defined. The time series voltage and spiking patterns will be recorded and saved by the model. These can either be used in future analyses or plotted via a raster plot.

User profile:
This package is geared towards individuals with no computational or neuroscience background as well as computational neuroscience researchers. The user must have a baseline knowledge of being able to download packages and use a Jupiter notebook or the command line to use this tool.

Use cases:

1. Interactive interface
    a. <u>Objective</u>: Use the toy model to understand how different network configurations provide different spiking dynamics
    b. <u>User-system interaction:</u> The user will simply run the build_network_interface() command to initialize the user interface. There, the layers, connectivity, stimulation layer, and stimulation intensity can be defined by the user. Additionally, a default connectivity button allows the user to auto populate these fields to a feasible network example. The user then pressed the "run simulation" button that progresses a loading bar. Once finished, the raster plot of spiking activity over time is shown and color coded by each layer.
2. Generation of neural networks
    a. <u>Objective</u>: Build a neural network of defined configuration that defines connections between each neuron and other neurons in the network
    b. <u>User-system interaction:</u> This function will be used by those with a higher understanding of neuroscience. In a jupyter notebook, the user will run the net = neural_net(...) command with desired inputs that define the structure of the network. The output is a net object that contains layers that contain neurons. A connectivity matrix of the network can be accessed from the net object that defines the connectivity from each neuron to the connected neurons.
3. Simulate neural activity over time

a. Objective: Much like use case 1, the goal is to understand neural dynamics in a network, but this interaction with the package gives more informative outputs than a simple plot such as voltages and spikes over time.
b. User-system interaction: This function should be used by those with a higher understanding of neuroscience and coding. In a jupyter notebook, the user will run the net, spikes, voltages = run_simulation(..) command with inputs that define the network structure and simulation structure. The outputs are a network of connections (net), the spikes over time (spikes), and voltages over time (voltages). The spikes and voltages are matrices of shape time x layer x neuron. In conjunction with the net object, both the neuron connectivity and functionality can be investigated.