This document explains all the details of the TextToNano project using an iOS app. I will first explain the python code then move on to the Swift code. Let's get started.

Micropython:

- we first import the necessary libraries
- we then define the pins we will be using, in this case the LCD pins are defines as follows:
    - ground to ground
    - red to vbus pin
    - sda to to pin A4 on arduino nano
    - scl to pin A5 on arduino nano
- we then initialize our wifi credentials as string variables:
    - ssid
    - password
- we then attempt to connect to the network:
    - station = network.WLAN(network.STA_IF) # Initialize the Wi-Fi interface station.active(True) # Activate the Wi-Fi interface station.connect(ssid, password) # Connect to the specified Wi-Fi network
- we wait until the connection is established:
    - # Wait until the connection is established while not station.isconnected(): pass # Continue looping until connectedprint("Connection successful") # Print a success message once connected print(station.ifconfig())
- we then set up the socket to listen on port 80:
    - # Setup the web server to listen on port 80 addr = socket.getaddrinfo('0.0.0.0', 80)[0][-1] # Get the address info for the socket s = socket.socket() # Create a socket object s.bind(addr) # Bind the socket to the address and port s.listen(1) # Start listening for incoming connections print('Listening on', addr) # Print the listening address
- we define a function to generate html page to be served
- in the main loop we handle incoming HTTP requests
    - we accept a new connection by defining the following variables:
        - conn, addr = s.accept() # Accept a new connection
    - we then receive the request and decode in into a string
    - we then extract the parameter from the http request and find the start of the text
    - we then find the end of the text
    - extract the text values and replace all the spaces
    - we then display the text on the LCD and if the string is longer than 16 it will move to the next line, this means that the max string to display is 32 chars.

Swift:

- we first define three variables, one to store the text that will be sent

    - one to store the serverIP to send the text to
    - one to store the status message of the response
- in the main body view we just create a simple text field and a button that calls the main function that does all the networking, this is all wrapped inside a v stack.
- we define the function sendTextToESP32() to send the text over wifi
    - we ensure the url is valid by doing a guard let
    - we create a url request with the specified url
    - we set the request to POST
    - set the content to content type
    - prepare the text to be sent by replacing the spaces
    - convert it to utf8
    - perform the http request and handle errors accordingly
    - start the network tasks with task.resume()

This project assumes that you have installed micropython and the LCD libraries required.