Using PMOD WIFI:

To use the PMOD Wi-Fi, most of the infrastructure is already available.

- For designs where performance is important, we found that Vivado v2015.4 worked significantly faster for the WIFI PMOD
- First, download the board files at https://reference.digilentinc.com/reference/software/vivado/board-files. Copy the appropriate board files into C:\Xilinx\Vivado\2016.2\data\boards\board_files.
  - I don't think this is super important, but it does make life easier
  - If you are using the school computers (highly recommended for speed reasons), you don't need to do this step, it is already done
- Next, download the Digilent PMOD IP's from https://github.com/Digilent/vivado-library/archive/master.zip and unzip it
- Start a new project and select the Nexys4 ddr board as your target device
- Add a Microblaze, UARTLite IP, and DDR by following the tutorials on piazza
  - The PMOD Wi-Fi requires more memory than we have BRAM, so pay attention to the line in the DDR tutorial that says "Enable Peripheral AXI Instruction Interface should be selected and the M_AXI_IP port should be connected to the memory slave"
- In Vivado, go to project settings, click the IP tab, click the add button, go to the unzipped Digilent IP folder and select
  - /vivado-library-master/if/pmod_v1_0
  - /vivado-library-master/ip/Pmods/Pmod_Bridge_v1_0
  - /vivado-library-master/ip/Pmods/PmodWIFI_v1_0
- Click the board tab in your block diagram, and if you have PMOD ports, select the PMOD port and add the WiFi Pmod IP, if not, just add the WiFi PMOD IP to your block design
- Run connection automation
- If the WiFi PMOD IP Pmod_out port is unconnected because Vivado is buggy and didn't give you the PMOD ports in the board options, just right click that port, click make it external, and in the constraints file connect it to the PMOD pins (they can be found at https://reference.digilentinc.com/reference/programmable-logic/nexys-4-ddr/reference-manual)
- I always connect WF_INTERRUPT to the Microblaze concat block, but I'm not sure whether you actually need to or not
- When you eventually open the SDK, you will find that in your hardware platform, you have drivers for the PMOD WiFi
- In the examples, they implement most of what you are likely to need (UDP client/server, TCP client/server and some others, but I ignored everything except the echo clients)
- If you copy the main.c from one of the example projects into your project you should be able to easily modify it for your needs. For HTTP use the TCP client example
  - You may have issues with relative paths in the header files, just find the appropriate header files and change the relative paths to absolute paths. Make sure you only have 1 copy of the file open, otherwise Xilinx SDK will just change the file back to the problematic version