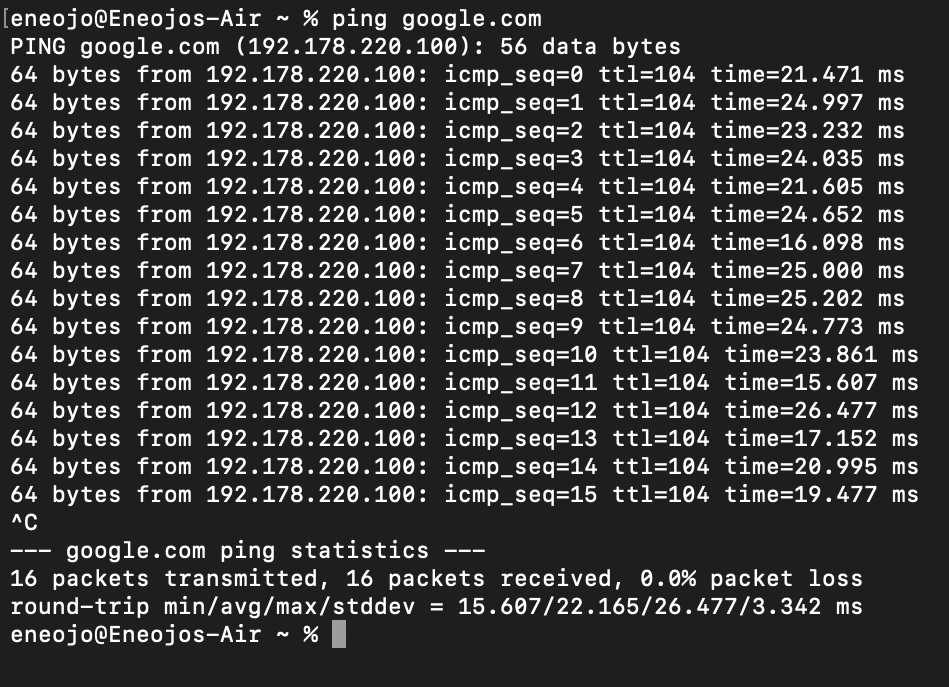
Necessary Knowledge

**PING** uses the ICMP protocol's echo request. It's basically a way to check if something is online by sending small packets. For example, running "ping google.com" sends packets from my network, and if all of them don’t come back, that means there’s packet loss. So if packets sent ≠ packets received, something’s off.

**Example:** Ping [google.com](http://google.com) sends a series of packets from my network over and until the command ends it keeps sending packets. At the end it measures the packet loss. If the packets sent and received aren't equal there is some form of packet loss.



**Application to the project**: I can use this to check if a website is online, responsive, or even real. It also hints if the connection is secure.

**Restrictions of ping**: The link has to be stripped down to just the domain. Also, some websites block it based on their network config.

**NSLLOOKUP:** A useful command for getting information from the DNS server. It gives you the IP address of domain names and instances of similar domain names

**Application to the project**: Helps get the website’s IP for other features I want to add.

**whois and who<IP>** It is able to bring out relevant contact information about a website. It can also bring out the region

**Restrictions of whois**: It does not work well on sites that are hosted through providers like vercel,render or heroku.

Some of these ideas were inspired by [Pingtool.org](http://pingtool.org)

The Main Idea of netshinobi

At its core netshinobi is able to strip information about a requested website that would be relevant to both developers and designers.

Core Features

*For Developers*- Detects tech stack including backend language and hosting  
- Scrapes the contact page  
- Checks if the site uses SSL  
- Shows where the site is hosted (country)

*For Designers*- Detects fonts and color palette  
- Grabs the favicon  
- Detects CSS frameworks like Tailwind or Bootstrap  
- Lets you view the site on mobile and desktop layouts

*For Both*- Gives a short AI-generated overview  
- Tells you if the site is currently online  
- Shows what kind of domain it is (.org, .com etc)

What I Need to Learn  
- Wireframing with Figma to plan layout  
- Stealth scraping using Cheerio and then Playwright for style detection  
- Better understanding of network APIs and Node modules like http and url -Better understanding of network APIs and Node modules like http and url

- Mongodb to check queries and possibly improve on the scraping of those individual sites

What I Learned  
- Got deeper into Puppeteer and Cheerio for scraping and automation  
- Realized Puppeteer docs aren’t great so I started building my own guide  
- Learned how Cloudflare works with hosting, CDNs, and security. It hides backend details using a reverse proxy  
- Understood DNS and HTTP protocols better  
- Learned what traits are common across frontend and backend frameworks

Biggest Challenge  
Detecting backend frameworks was tricky. Most are hidden behind Cloudflare, so I had to try unconventional and not-so-accurate methods.

ConclusionI think this project turned out pretty well and might actually be useful to others. I was able to hit the core features I wanted.