Website Health Checker

This is a simple command-line tool built in Go that checks whether a website is running on a specified domain and port. It can be useful for monitoring the uptime of websites or checking server availability.

Features:

- Check if a website is running by specifying the domain name and port.
- Default port is 80 (HTTP). You can also specify other ports like 443 (HTTPS).
- Includes a custom User-Agent header to avoid being blocked by servers that prevent automated requests.

Installation:

To use this tool, you need Go installed on your machine. Follow the instructions below to install and run the program.

Step 1: Clone the repository

git clone https://github.com/yourusername/website-health-checker.git

cd website-health-checker

Step 2: Install dependencies

This project uses the urfave/cli package. Install it by running:

go get github.com/urfave/cli/v2

Step 3: Run the application

To check the health of a website, use the following command:

go run main.go --domain example.com --port 80

Replace example.com with the domain you want to check and 80 with the port number.

Flags:

- --domain, -d : The domain name to check (Required).
- --port, -p : The port number to check (Optional, defaults to 80).

Example:

go run main.go --domain google.com --port 443

Output:

Status: 200 OK (200)

Expected Responses:

Some websites may respond differently based on security settings, bot protection, or rate limits. Below are some expected responses from different websites:

Website	Expected Response
example.com	200 OK or 404 Not Found
github.com	200 OK or rate-limited response
bbc.com	403 Forbidden if blocked
reddit.com	403 Forbidden or 429 Too Many Requests
wikipedia.org	200 OK or 403 Forbidden if restricted
twitter.com	403 Forbidden or 401 Unauthorized
amazon.com	403 Forbidden or 404 Not Found
google.com	200 OK but may challenge automated traffic
microsoft.com	200 OK but may challenge automated requests