Task: Website Crawler CLI Tool in Go

Overview:

Build a CLI tool in Go where a user can input a website URL, and the tool will:

1. Check if the site is reachable and return its HTTP status code.

2. Crawl the entire website (same domain) to generate a simple **site map**.

3. Print the status and site map in a readable format (and optionally export it as JSON or XML).

Requirements

Core Features

- Accept a URL as a command-line argument.

- Perform an HTTP GET request to the URL and return:

    - HTTP status code (e.g., 200, 404, 500).

    - Response time.

- Crawl the website:
    - Recursively find all internal links (same domain).
    - Ignore external links.
    - Avoid visiting the same URL more than once.
    - Respect robots.txt (basic support).
- Generate and print a **site map**:
    - Show page URLs in tree format or grouped by depth.
- Export site map as JSON (optional XML).

Bonus/Nice-to-Have

- Support concurrency (goroutines for crawling).

- Respect rate limiting (to not overload servers).

- Option to limit crawl depth.

- Add a `--json` or `--output` flag for exporting results.

Tech Stack

- Language: Go

- Libraries: `net/http`, `golang.org/x/net/html`, `cobra` or `urfave/cli` for CLI, `goquery` (optional)

Deliverables

- Source code in a public Git repo or zip file.

- `README.md` with:

    - Setup instructions.

    - Sample CLI usage.

    - Example output.

- Sample exported site map file.

Example CLI Usage

go run main.go https://example.com

Output:

[✓] Site: https://example.com

Status: 200 OK

Response Time: 120ms

Site Map:

- https://example.com

```
├── /about
├── /blog
│   ├── /blog/post-1
│   └── /blog/post-2
└── /contact
```

To export:

```
go run main.go https://example.com --json output.json
```