Website Cloning Using Go Language

Project REPORT submitted in partial fulfillment of the requirements

Submitted By

Motamarri Jaya Naga Venkata Sai (208W1A12A0)

Under The Guidance Of

M . Ramesh (Assistant Professor)

Prakash Kaja (CEO of Saadruso Company)

DEPARTMENT OF INFORMATION TECHNOLOGY

V R SIDDHARTHA ENGINEERING COLLEGE

(AUTONOMOUS - AFFILIATED TO JNTU-K, KAKINADA)

Approved by AICTE & Accredited by NBA

KANURU, VIJAYAWADA-520007

ACADEMIC YEAR

(2022-23)

V.R.SIDDHARTHA ENGINEERING COLLEGE

(Affiliated to JNTUK: Kakinada, Approved by AICTE, Autonomous)

(An ISO certified and NBA accredited institution)

Kanuru, Vijayawada – 520007

CERTIFICATE

This is to certify that this project report titled **"Go language"** is a bonafide record of work done by **M . J . N. V. Sai (208W1A12A0)** under my guidance and supervision is submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Information Technology, **V.R. Siddhartha Engineering College** (Autonomous under JNTUK) during the year 2022-23.

(M . Ramesh)
Assistant Professor
Officer
Dept. of Information Technology

(K . Prakash)
Chief Executive

Saadruso Company

ACKNOWLEDGEMENT

First and foremost, I sincerely salute our esteemed institution **V.R SIDDHARTHA ENGINEERING COLLEGE** for giving me this opportunity for fulfilling my project. I am grateful to our Principal **Dr. A.V.RATNA PRASAD**, for his encouragement and support all through the way of my project.

On the submission of this Project report, I would like to extend my honour to **Dr. M.Suneetha**, Head of the Department, IT for her constant motivation and support during the course of my work.

I feel glad to express my deep sense of gratefulness to my project guide **M.Ramesh, Assistant Professor** for **his/her** guidance and assistance in completing this project successfully.

I would also like to convey my sincere indebtedness to all faculty members, including supporting staff of the Department, friends and family members who bestowed their great effort and guidance at appropriate times without which it would have been very difficulty on my part to finish the project work.

Introduction :

Go is an open-source, statically typed, and compiled programming language designed by Rob Pike, Robert Griesemer, and Ken Thompson. The language, that appeared in the market in 2009, was designed with an intention to enhance programming productivity in the era of networked machines, multicore, and huge codebases. Something for which the Google team picked the best characteristics of the popular languages, like:

- Static typing and runtime efficiency of C++.
- Usability and Readability of Python and JavaScript.
- Object Oriented Programming (OOPs) concept ofSmalltalk.
- Concurrency element of Newsqueak.

Features of GO:

1. Open-Source

The foremost characteristic of Golang programming language is that it is open-source. That means, anyone can download and experiment with the code to bring better codes into picture and fix related bugs.

2. Static Typing

Go is a statically typed programming language and works with a mechanism that makes it possible to compile code accurately while taking care of type conversions and compatibility level. This gives developers freedom from challenges associated with dynamically typed languages.

3. Concurrency Support

One of the prime characteristics of go programming language is its concurrency support.

Golang, unlike other programming languages, offers easier and trackable concurrency options. This makes it easier for app developers to complete requests at a faster pace, free up allocated resources and network earlier, and much more.

4. Powerful Standard Library and Tool Set

This programming language also comes loaded with a robust standard library. This libraries offer ample components that gives developers an escape from turning towards third party packages anymore.

Also, it offers a wider range of tools that makes development process efficient. This includes:

- **Gofmt:** It automatically formats your Go code, which eventually brings a major impact on readability.
- **Gorun:** This tool is used to add a 'bang line' in the source code to run it, or run a similar sode code file explicitly. It is often used by Go developers when experimenting with codes written in Python.
- **Goget:** The Goget tool downloads libraries from GitHub and save it to your GoPath so that you can easily import the libraries in your app project.
- **Godoc:** The tool parses Go source code, including comments and creates a documentation in HTML or plain text format. The documentation made is tightly coupled with codes it documents and can be easily navigated with one click.
-

5. Testing Capabilities

Go language also offers an opportunity to write unit tests along with writing the app codes. Besides, it avails support to understand code coverage, benchmark tests, and write example codes to create your own code documentation.

ABSTRACT :

Go is a great language for creating simple yet efficient web servers and web services. It provides a built-in HTTP package that contains utilities for quickly creating a web or file server.

The goal of this tutorial is to create a web server that can accept a GET request and serve a response. We'll use the server to serve static files, acting as a file server. We'll then make the web server respond to a POST request coming from a form submission

Setup

You'll need Go version 1.11 or higher to follow this tutorial.

Create the following files index. And about.

SOFTWARE REQUIRMENTS OF THE PROJECT :

Any editor like Notepad and Visual Studio Code,

Google Go Compiler.

VS CODE

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including C#, Java, JavaScript, Go, Node.js, Python, C++, C, Rust and Fortran. It is based on the Electron framework,[20] which is used to develop Node.js web applications that run on the Blink layout engine. Visual Studio Code employs the same editor component (code named "Monaco") used in Azure DevOps (formerly called Visual Studio Online and Visual Studio Team Services). Out of the box, Visual Studio Code includes basic support for most common programming languages. This basic support includes syntax highlighting, bracket matching, code folding, and configurable snippets. Visual Studio Code also ships with IntelliSense for JavaScript, TypeScript, JSON, CSS, and HTML, as well as debugging support for Node.js. Support for additional languages can be provided by freely available extensions on the VS Code Marketplace.

CODE :

Main . go Program :

```
package main

import "github.com/imthaghost/goclone/cmd"

func main() {
	cmd.Execute()
}
```

Clone . go Program :

```
package cmd

import (
	"context"
	"fmt"
	"net/http"
	"net/http/cookiejar"
	"net/url"
	"strings"

	"os/exec"

	"github.com/imthaghost/goclone/pkg/crawler"
	"github.com/imthaghost/goclone/pkg/file"
	"github.com/imthaghost/goclone/pkg/html"
	"github.com/imthaghost/goclone/pkg/parser"
	"github.com/imthaghost/goclone/pkg/server"
)
```

```
// Clone the given site :)
func cloneSite(ctx context.Context, args []string) error {
	jar, err := cookiejar.New(&cookiejar.Options{})
	if err != nil {
		return err
	}
	var cs []*http.Cookie
	if len(Cookies) != 0 {
		cs = make([]*http.Cookie, 0, len(Cookies))
		for _, c := range Cookies {
			ff := strings.Fields(c)
			for _, f := range ff {
				var k, v string
				if i := strings.IndexByte(f, '='); i >= 0 {
					k, v = f[:i], strings.TrimRight(f[i+1:], ";")
				} else {
					return fmt.Errorf("No = in cookie %q", c)
				}
				cs = append(cs, &http.Cookie{Name: k, Value: v})
			}
		}
		for _, a := range args {
			u, err := url.Parse(a)
			if err != nil {
				return fmt.Errorf("%q: %w", a, err)
			}
			jar.SetCookies(&url.URL{Scheme: u.Scheme, User: u.User,
Host: u.Host}, cs)
		}
	}
```

```
	var firstProject string
	for _, u := range args {
		isValid, isValidDomain := parser.ValidateURL(u), parser.ValidateDomain(u)
		if !isValid && !isValidDomain {
			return fmt.Errorf("%q is not valid", u)
		}
		name := u
		if isValidDomain {
			u = parser.CreateURL(name)
		} else {
			name = parser.GetDomain(u)
		}
		projectPath := file.CreateProject(name)
		if firstProject == "" {
			firstProject = projectPath
		}

		if err := crawler.Crawl(ctx, u, projectPath, jar, ProxyString, UserAgent); err != nil {
			return fmt.Errorf("%q: %w", u, err)
		}
		// Restructure html
		if err := html.LinkRestructure(projectPath); err != nil {
			return fmt.Errorf("%q: %w", projectPath, err)
		}

	}
	if Serve {
		cmd := exec.CommandContext(ctx, "open", "http://localhost:5000")
		if err := cmd.Start(); err != nil {
```

```
                return fmt.Errorf("%v: %w", cmd.Args, err)
            }
            return server.Serve(firstProject)
        } else if Open {
            // automatically open project
            cmd := exec.CommandContext(ctx, "open", firstProject+"/index.html")
            if err := cmd.Start(); err != nil {
                return fmt.Errorf("%v: %w", cmd.Args, err)
            }
        }
        return nil
}
```

Root . go Program :

```
package cmd

import (
        "context"
        "log"
        "os"
        "os/signal"

        "github.com/spf13/cobra"
)

var (
        Open        bool
        Serve       bool
        UserAgent   string
        ProxyString string
```

```
	Cookies     []string


	// Root cmd
	rootCmd = &cobra.Command{
		Use:   "goclone <url>",
		Short: "Clone a website with ease!",
		Long:  `Copy websites to your computer! goclone is a utility that allows you to download a website from the Internet to a local directory. Get html, css, js, images, and other files from the server to your computer. goclone arranges the original site's relative link-structure. Simply open a page of the "mirrored" website in your browser, and you can browse the site from link to link, as if you were viewing it online.`,
		// TODO Update link once we change repo name
		Args:  cobra.ArbitraryArgs,
		Run: func(cmd *cobra.Command, args []string) {
			// Print the usage if no args are passed in :)
			if len(args) < 1 {
				if err := cmd.Usage(); err != nil {
					log.Fatal(err)
				}

				return
			}

			ctx, stop := signal.NotifyContext(context.Background(), os.Interrupt)
			defer stop()
			// Otherwise.. clone ahead!
			if err := cloneSite(ctx, args); err != nil {
				log.Fatalf("%+v", err)
			}
		},
	}
)
```

```
// Execute the clone command
func Execute() {
	// Persistent Flags
	pf := rootCmd.PersistentFlags()
	pf.BoolVarP(&Open, "open", "o", false, "Automatically open project in deafult browser")
	// rootCmd.PersistentFlags().BoolVarP(&Login, "login", "l", false, "Wether to use a username or password")
	pf.BoolVarP(&Serve, "serve", "s", false, "Serve the generated files using Echo.")
	pf.StringVarP(&ProxyString, "proxy_string", "p", "", "Proxy connection string. Support http and socks5 https://pkg.go.dev/github.com/gocolly/colly#Collector.SetProxy")
	pf.StringVarP(&UserAgent, "user_agent", "u", "", "Custom User Agent")
	rootCmd.Flags().StringSliceVarP(&Cookies, "cookie", "C", nil, "Pre-set these cookies")

	// Execute the command :)
	if err := rootCmd.Execute(); err != nil {
		log.Fatal(err)
	}
}
```

HTML . go Program :

```
package crawler

import (
	"fmt"
	"io/ioutil"
	"net/http"
	"os"
	"crypto/tls"
```

```
)

// HTMLExtractor ...
func HTMLExtractor(link string, projectPath string) {
	fmt.Println("Extracting --> ", link)

	http.DefaultTransport.(*http.Transport).TLSClientConfig = &tls.Config{InsecureSkipVerify: true}
	// get the html body
	resp, err := http.Get(link)
	if err != nil {
		panic(err)
	}

	// Close the body once everything else is compled
	defer resp.Body.Close()

	// get the project name and path we use the path to
	f, err := os.OpenFile(projectPath+"/"+"index.html", os.O_RDWR|os.O_CREATE, 0777)
	if err != nil {
		panic(err)
	}
	defer f.Close()
	htmlData, err := ioutil.ReadAll(resp.Body)

	if err != nil {
		panic(err)
	}
	f.Write(htmlData)
```

```
}
```

Write . go Program :

```
package file

import (
	"log"
	"os"
)

// CreateProject initializes the project directory and returns the path to the project
// TODO make function more modular to obtain different html files
func CreateProject(projectName string) string {
	// current workin directory
	path := currentDirectory()

	// define project path
	projectPath := path + "/" + projectName

	// create base directory
	err := os.MkdirAll(projectPath, 0777)
	check(err)

	// create CSS/JS/Image directories
	createCSS(projectPath)
	createJS(projectPath)
	createIMG(projectPath)

	// main inedx file
	_, err = os.Create(projectPath + "/" + "index.html")
```

```
	check(err)
	// project path
	return projectPath
}

// currentDirectory get the current working directory
func currentDirectory() string {
	path, err := os.Getwd()
	check(err)
	return path
}

// createCSS create a css directory in the current path
func createCSS(path string) {
	// create css directory
	err := os.MkdirAll(path+"/"+"css", 0777)
	check(err)
}

// createJS create a JS directory in the current path
func createJS(path string) {
	err := os.MkdirAll(path+"/"+"js", 0777)
	check(err)
}

// createIMG create a image directory in the current path
func createIMG(path string) {
	err := os.MkdirAll(path+"/"+"imgs", 0777)
	check(err)
}
```

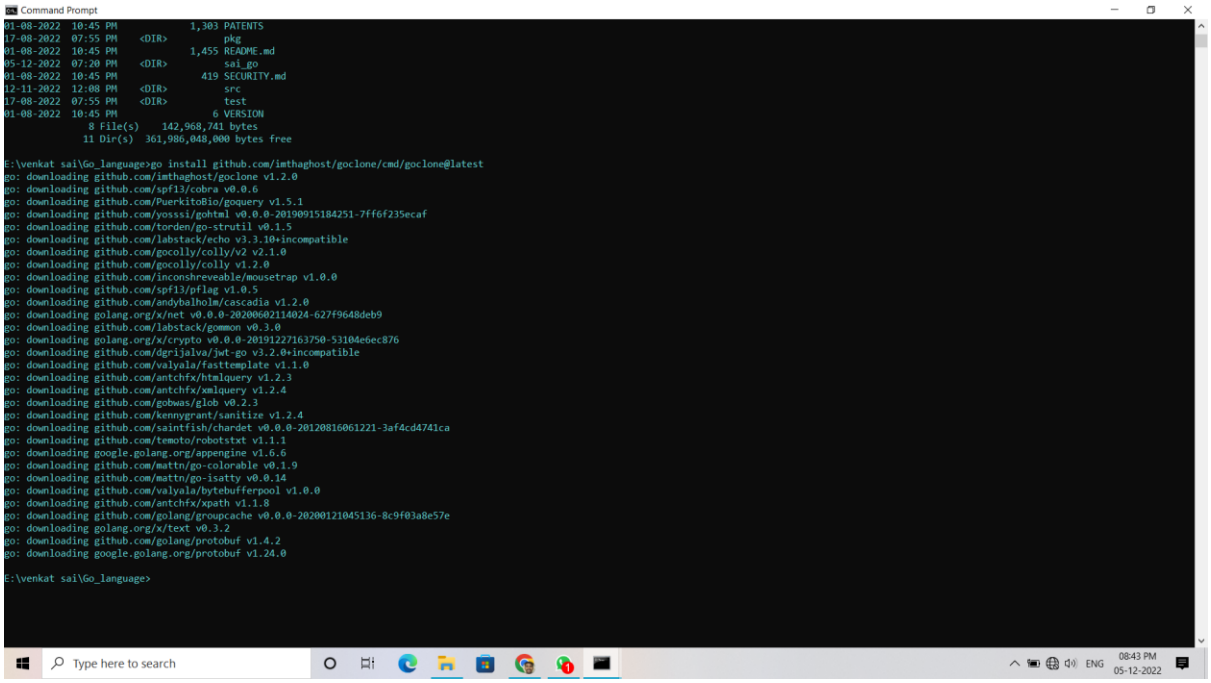
```
func check(err error) {
	if err != nil {
		log.Println(err)
	}
}
```

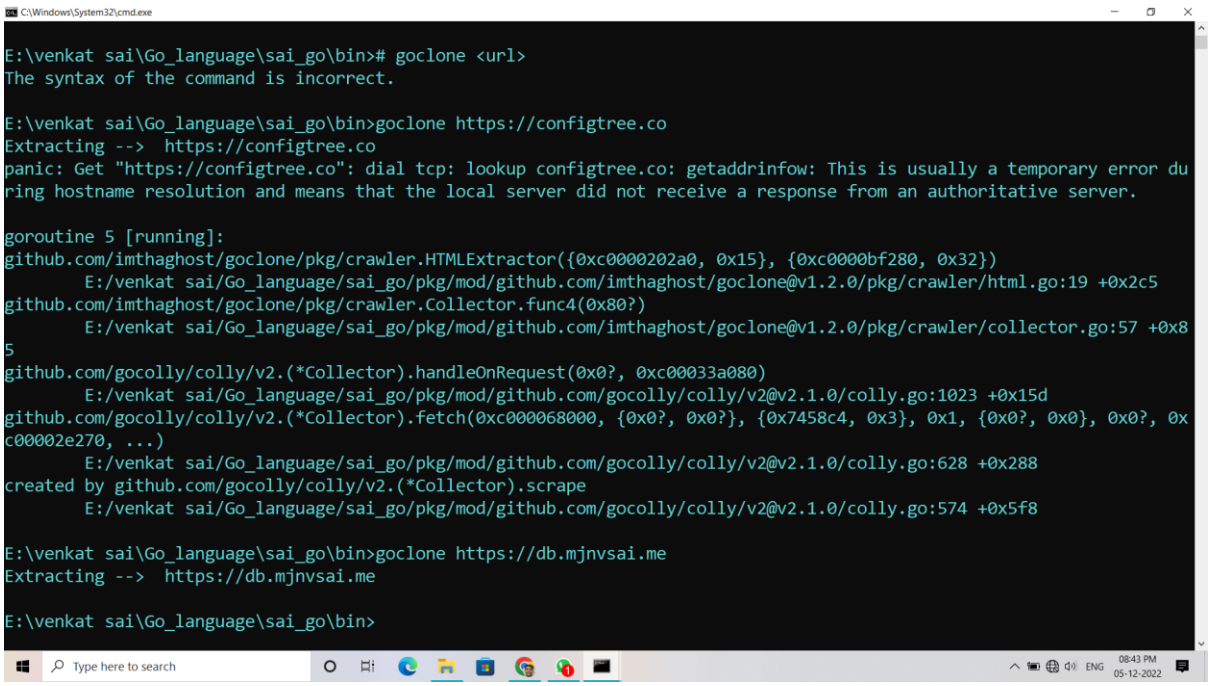
Server . go Program :

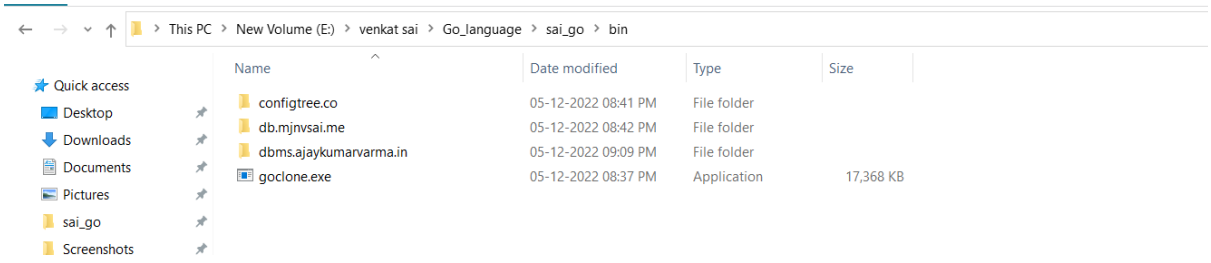
```
package server

import (
	"github.com/labstack/echo"
	"github.com/labstack/echo/middleware"
)

// Serve ...
func Serve(projectPath string) error {
	e := echo.New()
	// Log Output
	e.Use(middleware.Logger())
	e.Use(middleware.Recover())
	// CORS
	e.Use(middleware.CORSWithConfig(middleware.CORSConfig{
		AllowOrigins: []string{"*"},
		AllowMethods: []string{echo.GET, echo.HEAD, echo.PUT, echo.PATCH, echo.POST, echo.DELETE},
	}))
	// static files
	e.Static("/", projectPath)
	e.File("/", projectPath)

	return e.Start(":5000")
```

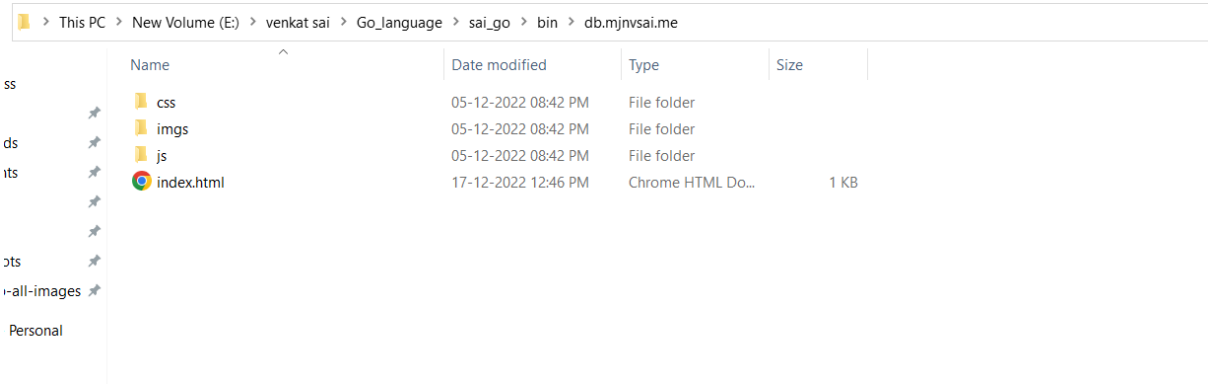
}

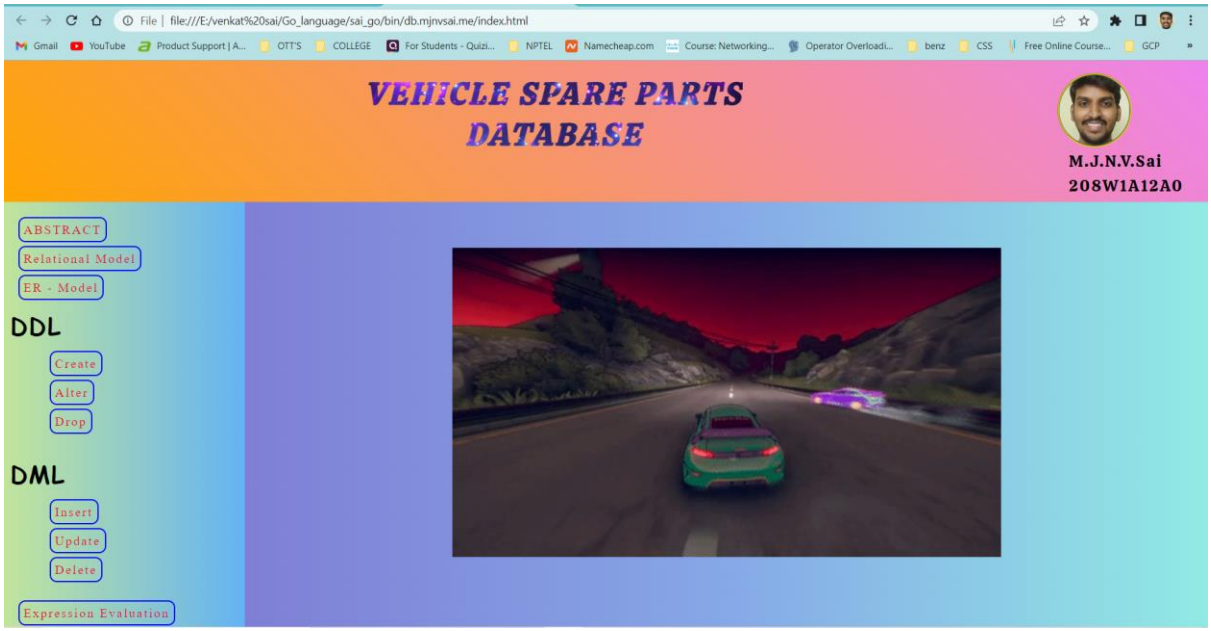
OUTPUT:


Command Prompt
01-08-2022 10:45 PM 1,303 PATENTS
17-08-2022 07:55 PM <DIR> pkg
01-08-2022 10:45 PM 1,455 README.md
05-12-2022 07:20 PM <DIR> sai_go
01-08-2022 10:45 PM 419 SECURITY.md
12-11-2022 12:08 PM <DIR> src
17-08-2022 07:55 PM <DIR> test
01-08-2022 10:45 PM 6 VERSION
8 File(s) 142,968,741 bytes
11 Dir(s) 361,986,048,000 bytes free
E:\venkat sai\Go_language>go install github.com/imthaghost/goclone/cmd/goclone@latest
go: downloading github.com/imthaghost/goclone v1.2.0
go: downloading github.com/spf13/cobra v0.0.6
go: downloading github.com/PuerkitoBio/goquery v1.5.1
go: downloading github.com/yosssi/gohtml v0.0.0-20190915184251-7ff6f235ecaf
go: downloading github.com/torden/go-strutil v0.1.5
go: downloading github.com/labstack/echo v3.3.10+incompatible
go: downloading github.com/gocolly/colly/v2 v2.1.0
go: downloading github.com/gocolly/colly v1.2.0
go: downloading github.com/inconshreveable/mousetrap v1.0.0
go: downloading github.com/spf13/pflag v1.0.5
go: downloading github.com/andybalholm/cascadia v1.2.0
go: downloading golang.org/x/net v0.0.0-20200602114024-627f9648deb9
go: downloading github.com/labstack/gommon v0.3.0
go: downloading golang.org/x/crypto v0.0.0-20191227163750-53104e6ec876
go: downloading github.com/dgrijalva/jwt-go v3.2.0+incompatible
go: downloading github.com/valyala/fasttemplate v1.1.0
go: downloading github.com/antchfx/htmlquery v1.2.3
go: downloading github.com/antchfx/xmlquery v1.2.4
go: downloading github.com/gobwas/glob v0.2.3
go: downloading github.com/kennygrant/sanitize v1.2.4
go: downloading github.com/saintfish/chardet v0.0.0-20120816061221-3af4cd4741ca
go: downloading github.com/temoto/robotstxt v1.1.1
go: downloading google.golang.org/appengine v1.6.6
go: downloading github.com/mattn/go-colorable v0.1.9
go: downloading github.com/mattn/go-isatty v0.0.14
go: downloading github.com/valyala/bytebufferpool v1.0.0
go: downloading github.com/antchfx/xpath v1.1.8
go: downloading github.com/golang/groupcache v0.0.0-20200121045136-8c9f03a8e57e
go: downloading golang.org/x/text v0.3.2
go: downloading github.com/golang/protobuf v1.4.2
go: downloading google.golang.org/protobuf v1.24.0
E:\venkat sai\Go_language>
Type here to search
ENG
08:43 PM
05-12-2022


C:\Windows\System32\cmd.exe
E:\venkat sai\Go_language\sai_go\bin># goclone <url>
The syntax of the command is incorrect.
E:\venkat sai\Go_language\sai_go\bin>goclone https://configtree.co
Extracting --> https://configtree.co
panic: Get "https://configtree.co": dial tcp: lookup configtree.co: getaddrinfow: This is usually a temporary error du
ring hostname resolution and means that the local server did not receive a response from an authoritative server.
goroutine 5 [running]:
github.com/imthaghost/goclone/pkg/crawler.HTMLExtractor({0xc0000202a0, 0x15}, {0xc0000bf280, 0x32})
E:/venkat sai/Go_language/sai_go/pkg/mod/github.com/imthaghost/goclone@v1.2.0/pkg/crawler/html.go:19 +0x2c5
github.com/imthaghost/goclone/pkg/crawler.Collector.func4(0x80?)
E:/venkat sai/Go_language/sai_go/pkg/mod/github.com/imthaghost/goclone@v1.2.0/pkg/crawler/collector.go:57 +0x8
5
github.com/gocolly/colly/v2.(*Collector).handleOnRequest(0x0?, 0xc00033a080)
E:/venkat sai/Go_language/sai_go/pkg/mod/github.com/gocolly/colly/v2@v2.1.0/colly.go:1023 +0x15d
github.com/gocolly/colly/v2.(*Collector).fetch(0xc000068000, {0x0?, 0x0?}, {0x7458c4, 0x3}, 0x1, {0x0?, 0x0}, 0x0?, 0x
c00002e270, ...)
E:/venkat sai/Go_language/sai_go/pkg/mod/github.com/gocolly/colly/v2@v2.1.0/colly.go:628 +0x288
created by github.com/gocolly/colly/v2.(*Collector).scrape
E:/venkat sai/Go_language/sai_go/pkg/mod/github.com/gocolly/colly/v2@v2.1.0/colly.go:574 +0x5f8
E:\venkat sai\Go_language\sai_go\bin>goclone https://db.mjnvsai.me
Extracting --> https://db.mjnvsai.me
E:\venkat sai\Go_language\sai_go\bin>
Type here to search
ENG
08:43 PM
05-12-2022


This PC › New Volume (E:) › venkat sai › Go_language › sai_go › bin
Quick access
Desktop
Downloads
Documents
Pictures
sai_go
Screenshots
Name
Date modified
Type
Size
configtree.co
05-12-2022 08:41 PM
File folder
db.mjnvsai.me
05-12-2022 08:42 PM
File folder
dbms.ajaykumarvarma.in
05-12-2022 09:09 PM
File folder
goclone.exe
05-12-2022 08:37 PM
Application
17,368 KB


This PC › New Volume (E:) › venkat sai › Go_language › sai_go › bin › db.mjnvsai.me
Name
Date modified
Type
Size
css
05-12-2022 08:42 PM
File folder
imgs
05-12-2022 08:42 PM
File folder
js
05-12-2022 08:42 PM
File folder
index.html
17-12-2022 12:46 PM
Chrome HTML Do...
1 KB
Personal


VEHICLE SPARE PARTS DATABASE
M.J.N.V.Sai
208W1A12A0
ABSTRACT
Relational Model
ER - Model
DDL
Create
Alter
Drop
DML
Insert
Update
Delete
Expression Evaluation

Conclusion :

In this project we learn the creating a built-in web server task. But with the help of some basic knowledge of GO programming we can create a basic application for user and it will help them to to build a web server in any place easily.