To become a proficient full-stack developer with Go (Golang), you need to be well-versed in both front-end and back-end development, as well as the specific features and best practices of Go. Here's a comprehensive guide to the skills and knowledge you should acquire:

Go (Golang) Basics
1. **Syntax and Data Types**
   - Variables and constants
   - Basic types (int, float, string, bool, etc.)
   - Complex types (structs, arrays, slices, maps)

2. **Control Structures**
   - Conditional statements (if-else, switch-case)
   - Loops (for, range)
   - Error handling

3. **Functions**
   - Function declarations and returns
   - Variadic functions
   - Anonymous functions and closures
   - Methods and interfaces

4. **Concurrency**
   - Goroutines
   - Channels (buffered and unbuffered)
   - Select statement
   - Sync package (Mutex, WaitGroup, etc.)

5. **File Handling**
   - Reading and writing files
   - Working with directories
   - File I/O operations

6. **Packages and Modules**
   - Importing packages
   - Creating and managing modules
   - Standard library usage

Back-End Development with Go
1. **Building Web Servers**
   - net/http package for building HTTP servers
   - Handling routes and requests
   - Middleware (custom and third-party)

2. **API Development**
   - RESTful API design
   - JSON encoding and decoding
   - Handling query parameters, headers, and forms
   - Authentication and authorization (JWT, OAuth)

3. **Database Integration**
   - SQL databases (PostgreSQL, MySQL) using database/sql package
   - ORMs like GORM
   - NoSQL databases (MongoDB) using third-party packages

4. **Templating**
   - html/template and text/template packages
   - Creating and rendering templates

5. **Testing**
   - Writing unit tests (testing package)
   - Writing integration tests
   - Using testing frameworks (Testify)

Front-End Development
1. **HTML/CSS**
   - Basic HTML and CSS
   - CSS frameworks (Bootstrap, Tailwind CSS)
   - Responsive design (media queries, mobile-first design)

2. **JavaScript**
   - Basic syntax and operations
   - DOM manipulation
   - Event handling
   - Fetch API for making HTTP requests

3. **Front-End Frameworks**
   - React (components, state, props, lifecycle methods, hooks)
   - Vue or Angular (optional but beneficial)
   - State management (Redux, Context API for React)

4. **Build Tools**
   - Module bundlers (Webpack, Parcel)
   - Package management (npm, Yarn)

5. **Routing**
   - React Router or similar libraries

Additional Skills
1. **Version Control**
   - Git (basic commands, branching, merging, pull requests)
   - Platforms (GitHub, GitLab, Bitbucket)

2. **CI/CD**
   - Continuous Integration and Continuous Deployment basics
   - Tools like Jenkins, Travis CI, GitHub Actions

3. **Web Security**
   - Understanding common web security issues (XSS, CSRF, SQL Injection)
   - Best practices for securing web applications

4. **Performance Optimization**
   - Techniques for optimizing back-end performance in Go
   - Front-end optimization techniques

5. **DevOps Basics**
   - Containerization (Docker)
   - Orchestration (Kubernetes)
   - Cloud services (AWS, Google Cloud, Azure)

Development Environment and Tools
1. **Code Editors and IDEs**
   - Visual Studio Code, GoLand
   - Debugging tools and techniques

2. **Package Management**
   - Go modules (go mod)

3. **Linters and Formatters**
   - gofmt, golint, staticcheck

By mastering these concepts, you'll be well-equipped to handle full-stack development tasks using Go (Golang).