**SPL TERM PROJECT REPORT**

|  |  |
| --- | --- |
| **Name** | **Email ID** |
| Ankita Singh | asingh63@uncc.edu |
| Sneh Jain | sjain42@uncc.edu |
| Varun Jain | vjain10@uncc.edu |
| Dhruvil Patel | dpate172@uncc.edu |
| Mounica Arendala | marendal@uncc.edu |
| Bhargavi Lata Mugata | [bmugata@uncc.edu](mailto:bmugata@uncc.edu) |

**Introduction - Motivation**

The reason behind developing this application is According to a research, in an eight-hour workday, the average worker is only productive for two hours and 53 minutes. Yep, in an 8-hour workday, we only end up working for a maximum of 3 hours on average. It’s a disaster!

So we end up searching on the internet- from productivity books to techniques, to meditation-trying our hand at everything under the sun to be more productive. While looking for newer ways and techniques to improve productivity, we often forget the age-old trick to get work done quickly- **a to-do list**.

This application helps you prioritize your task and gives you an action plan for the day. In this way, you always know what you need to do.

We used Go-Lang for our project due to the fact that It is a compiled language, fast and high-performance language intended to be simple and is designed to be easy to read and understand.

**Advantages**

Golang has indisputable architectural advantages over its predecessors and has Characteristics like Open-Source,Static typing,Concurrency support,Powerful Standard Library and Tool Set,Golang is an opinionated language, has Garbage Collection And Testing Capabilities.

Go was created at Google by Rob Pike, Robert Griesemer and Ken Thompson, and it first appeared in Nov 2009. Go was invented at such a time when multicore CPU architectures were common everywhere, and no programming language simplified the development of multithreaded applications.

Go as a language is more similar to C, however in addition to C features, Go offers memory safety, garbage collection, structural typing, and CSP-Style concurrency.

**Limitations**

The limitations of Go are Runtime safety is not that good

Absence of manual memory management (lead to overhead garbage collection).

Lack of Flexibility: This statically typed language offers less flexibility of coding, as per the opinion of those advocating dynamically typed languages.

Lack of Generics in the language.

**Problem Statement:**

We are building a To-Do List App in Golang. The To-Do list app allows users to add the to-do items and can edit and delete it later.

Managing tasks in everyday life is very tough to do these days. People end up wasting time due to mismanagement of their daily to daily tasks. Therefore, we are coming up with a solution as a simple To-do list.

**Technology:**

* **Server Handling:** Go
* **Database:** MySQL
* **Front end:** HTML, CSS and bootstrap.
* **Version control:** GitHub

**Proposed solution if any**

We’ll build a classic and simple Golang API server that connects to a front-end page

Our Golang API server will use:

* MySQL as our database
* GORM as an ORM to interact with our database
* Request router using gorilla/mux
* Logrus for logging

**Intuition:**

If somehow we can keep track of our tasks digitally, then we can manage our time efficiently. So, here comes an idea to develop a todo list, which can keep track of our tasks and also prioritize the tasks. With the use of Golang and the features it becomes easier to create it.

Currently, there are many counterparts of our projects out there in the market. But, none of it solves the problem by handling concurrency. Concurrency controls help many users to perform crud operations in it simultaneously.

**Description of Algorithms:**

CRUD Meaning: *CRUD* is an acronym that comes from the world of computer programming and refers to the four functions that are considered necessary to implement a persistent storage application: *create, read, update* and *delete*. Persistent storage refers to any data storage device that retains power after the device is powered off, such as a hard disk or a solid-state drive. In contrast, random access memory and internal caching are two examples of volatile memory - they contain data that will be erased when they lose power.

### **Explaining CRUD Operations**

Organizations that keep track of customer data, accounts, payment information, health data, and other records require data storage hardware and applications that provide persistent storage. This data is typically organized into a database, which is simply an organized collection of data that may be viewed electronically. There are many types of databases: hierarchical databases, graph databases, and object-oriented databases to name a few. The most commonly implemented type of database is a *relational database*, which consists of data tabled in rows and columns and connected to other tables with complementary information by a system of keywords that includes *primary keys and* *foreign keys*.

### **CRUD Databases and the Applications to Manage Them**

The *CRUD* acronym identifies all of the major functions that are inherent to *relational databases* and the applications used to manage them, which include Oracle Database, Microsoft SQL Server, MySQL, and others.

**Discuss any challenges that were not anticipated**

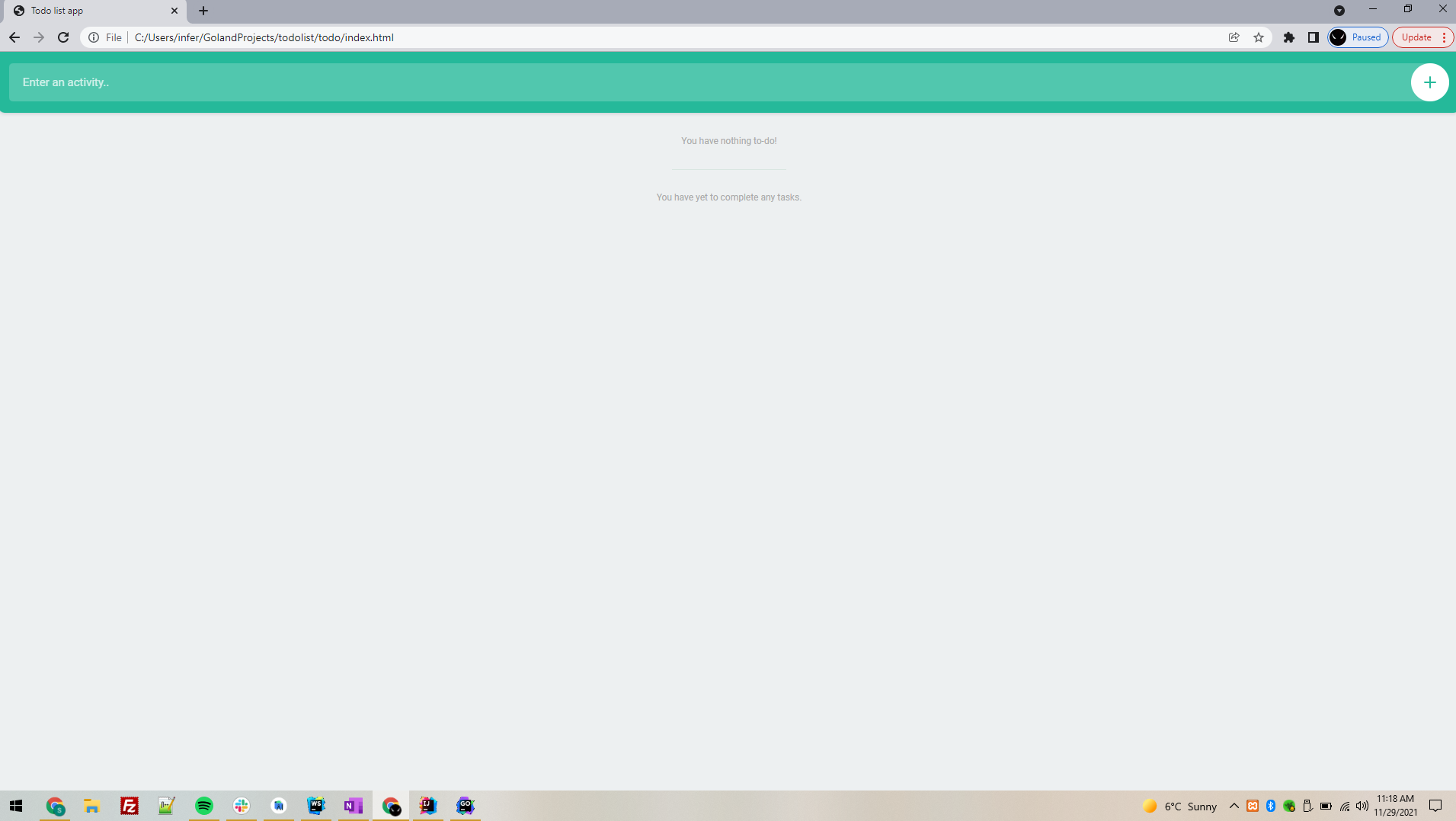
* Understanding the Golang as all the group members are working on the language for the first time.
* Learning all the packages and syntax to achieve our project goal.
* Setting up the basic structure of the project.

**Experiments**

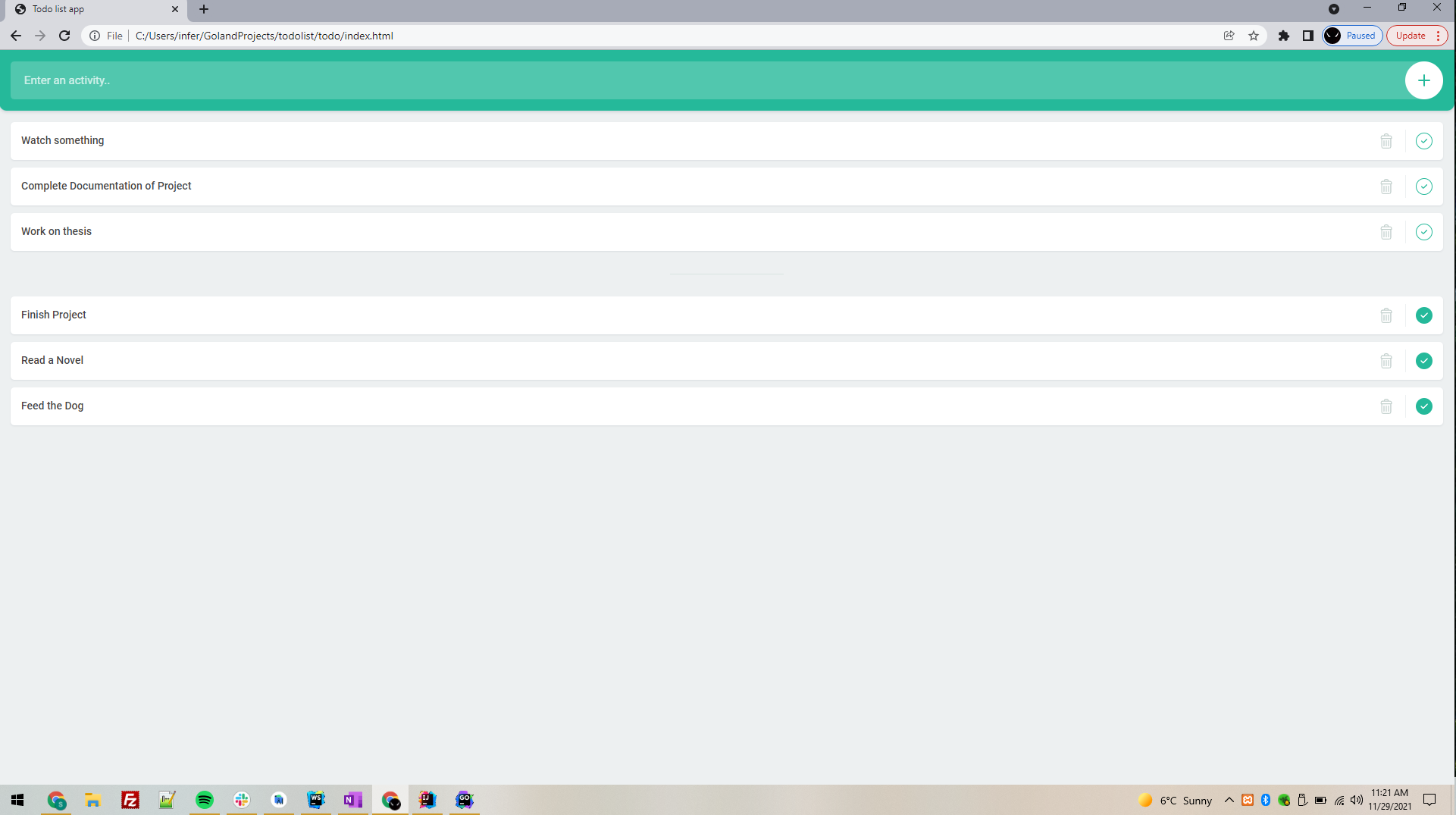
The use cases in our project can be an assigned task or to-do work which is not done or is completed, The tasks can be deleted or completed.

The list of tasks is handled which dynamically handles those tasks completed and those are yet to be done.

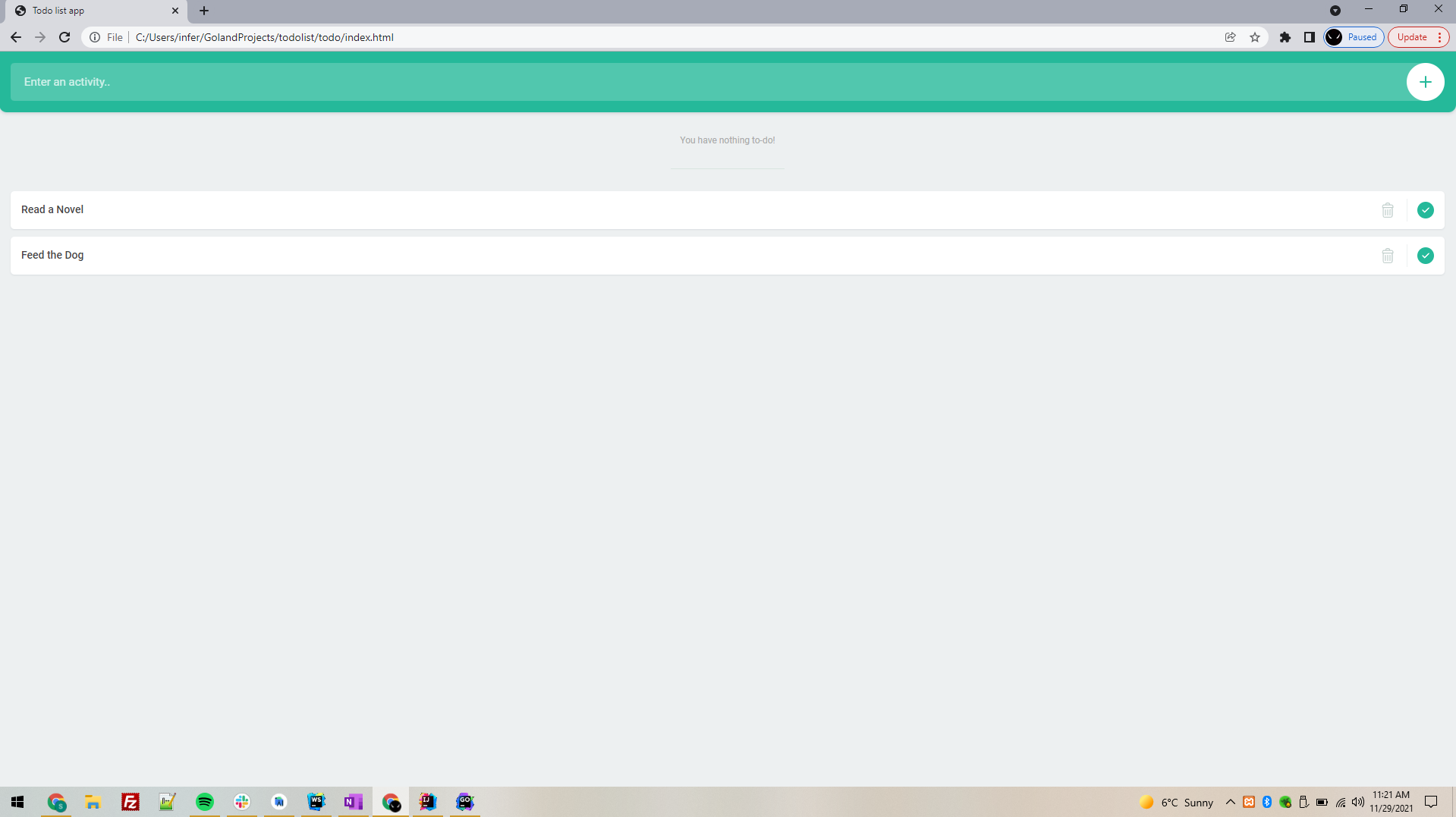
1. No assigned task



1. Task yet to be completed and completed task in the list



1. Empty to do tasks and completed tasks list



**Future enhancements:**

* Build the feature to add reminders for the To-Do tasks.
* Integrate the To-Do List app with Google calendar

**Conclusions**

Every programmer has their own preferences. Many people who are currently utilizing Python or Java are hesitant to make the leap. Furthermore, there are many who believe that Golang will never be able to replace languages as versatile as C# or Java.

However, the developer's hesitation to use Golang is more commonly owing to the fact that he or she has already spent months or years learning and mastering more sophisticated languages. Switching to Golang would entail giving up knowledge that had taken a long time to gain.

However, we must acknowledge that Golang has flaws and is not without flaws. As a result, only engineers who have worked with a variety of languages can objectively compare Golang to the top programming languages. Furthermore, it is dependent on your individual requirements; Golang may be sufficient for some, but not for others.