***Arpit Singh***

***19BCG10069***

***Appointy – Task 1***

***Technical Task***

***(Internship)***

**Steps:**

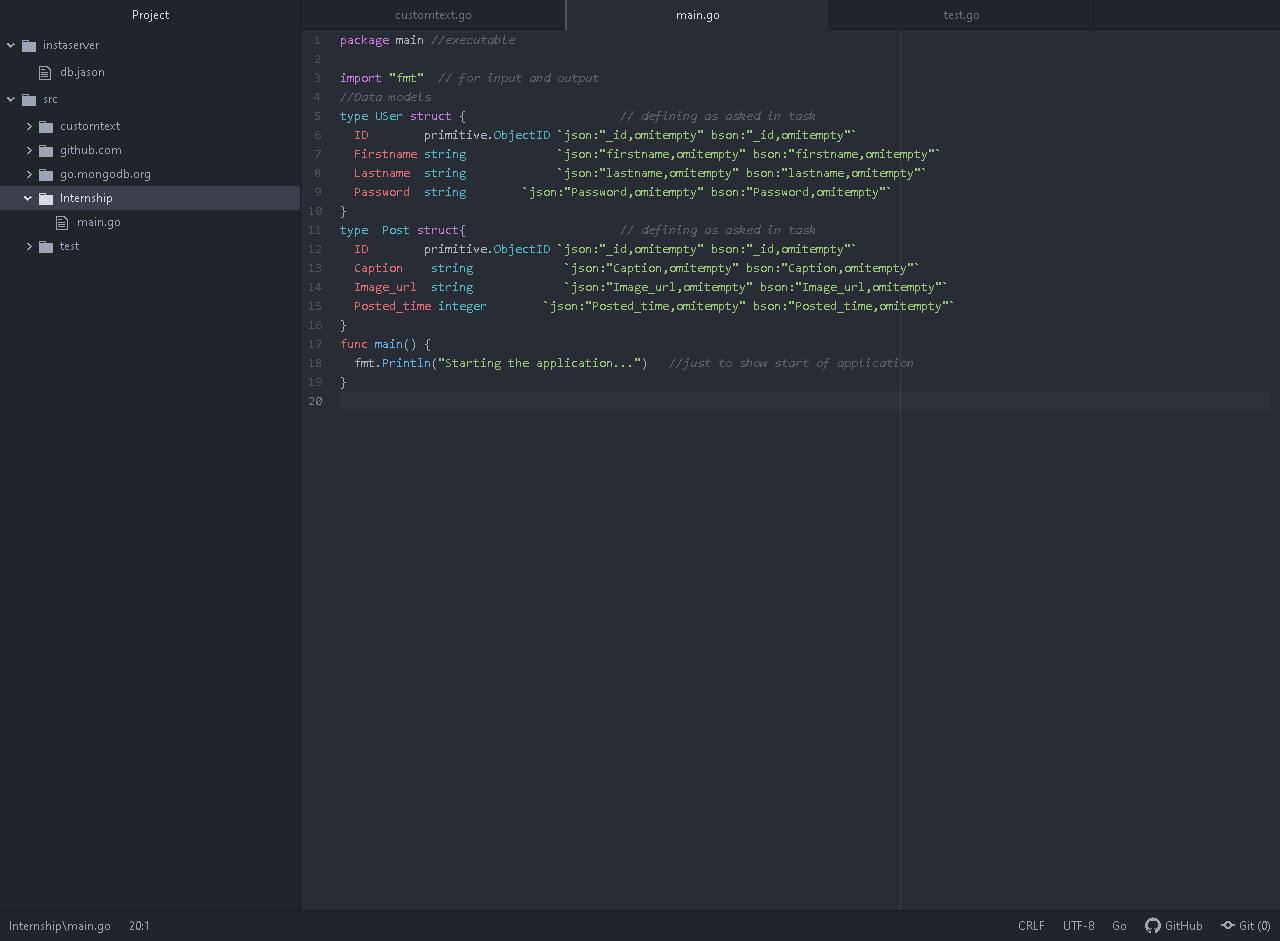
1. Installing Golang and Setting up the Environment
   1. Install(Setup for windows) golang from Golang.org
   2. Setup the System (Environment) variable as per the desired directory
      1. Use commands: go env; Check GoRoot and GoPath specifically.
   3. For Example: To set workstation as a folder in Desktop set env variable as: *C:\Users\ArpitSG\Desktop\Go-Workspace.*
2. Installing MongoDB and Setting up the Environment
   1. Install(Setup for windows) MongoDB from MongoDB.com( community server )
   2. Setup the System (Environment) variable as per the desired directory
   3. For Example: *C:\Program Files\MongoDB\Server\5.0\bin.*
3. Setup a Connection between MongoDB cluster(Server) and Golang(file)
   1. Import dependencies :
      1. go get github.com/gorilla/mux (For Managing htttp requests)
      2. go get go.mongodb.org/mongo-driver/mongo(For Managing and establishing connection with MongoDB)

**Coding**

Create a file under GOPath Directory

(C:\Users\ArpitSG\Desktop\Go-Workspace\src\Internship\main.go)

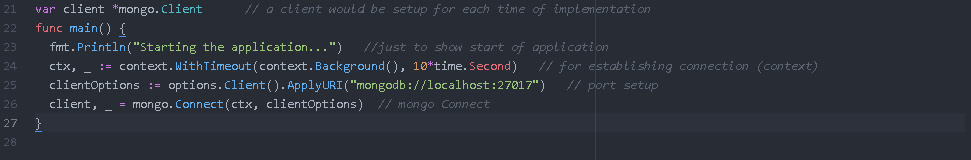
Open a text Editor (Atom) and write the following code that just puts in basic data model and outline of our go file.



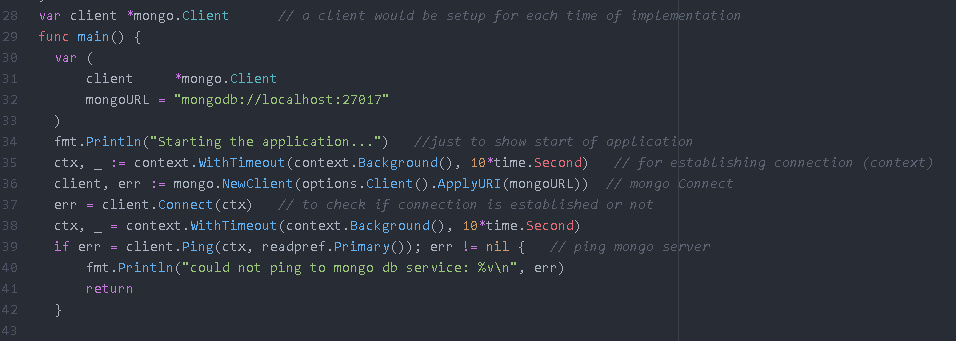
*Json – for user understanding (Web client)*

*Bson – for MongoDB interpretation*

Next lets create a mongo client and establish connection(within main function along with context time) each time a function of our application is implemented.



Next lets define the router to complete the connection process with MongoDB by passing in our router and port.



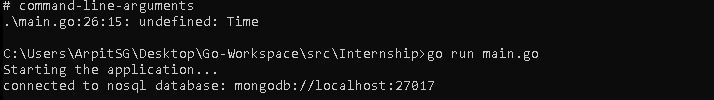
Now lets Check everything is working properly.

To do so lets go to terminal (Cmd) and execute are main.go (file) with:

(go run main.go)

Note you should be in the GoPath file directory

(C:\Users\ArpitSG\Desktop\Go-Workspace\src\Internship\main.go)



Now with this the connection to server is setup and we have how the basic data model of our application looks like.

***­Next lets perform the task as descirebed in the doc file:***

*So in the folder named TheMainAPI all the required coding tasks are done.*

***\*NOTE: Proper comments are added in each file for better understanding.\****

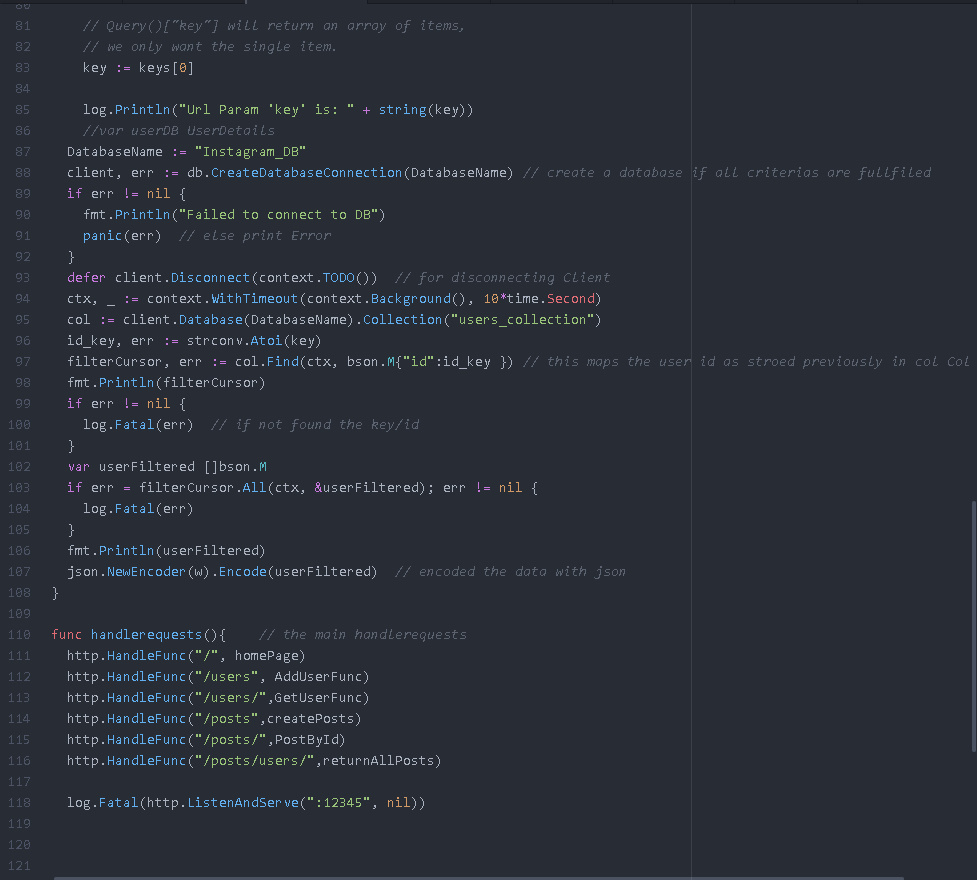
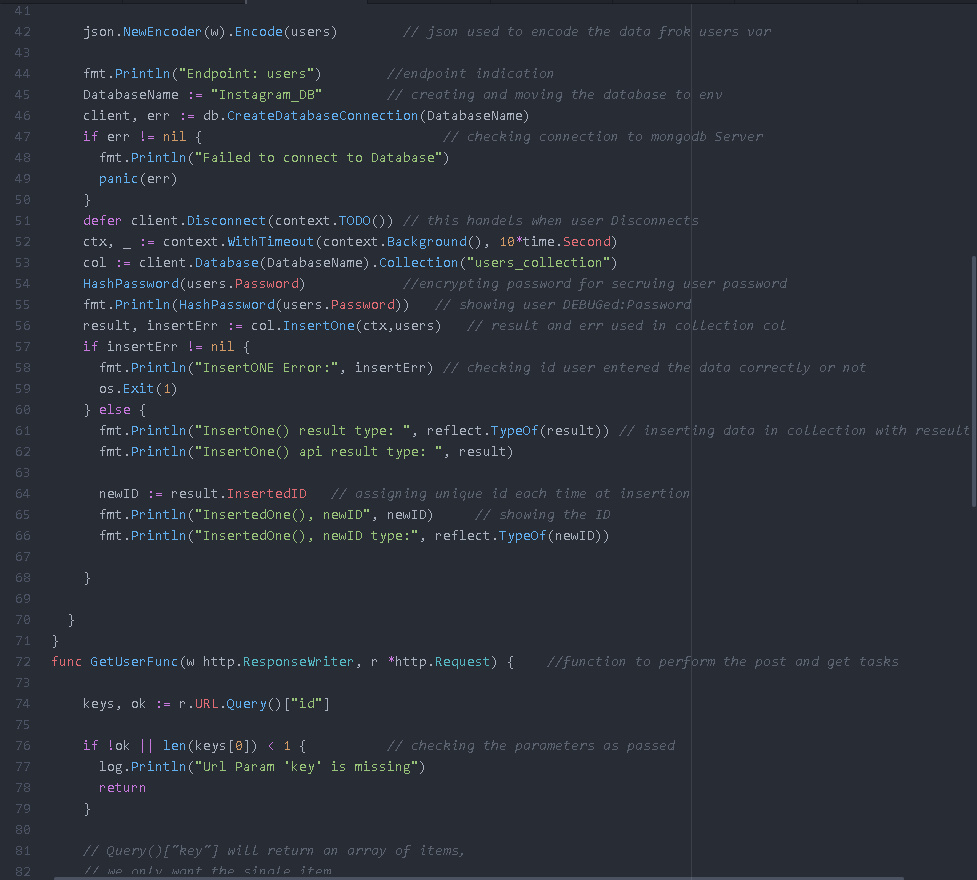
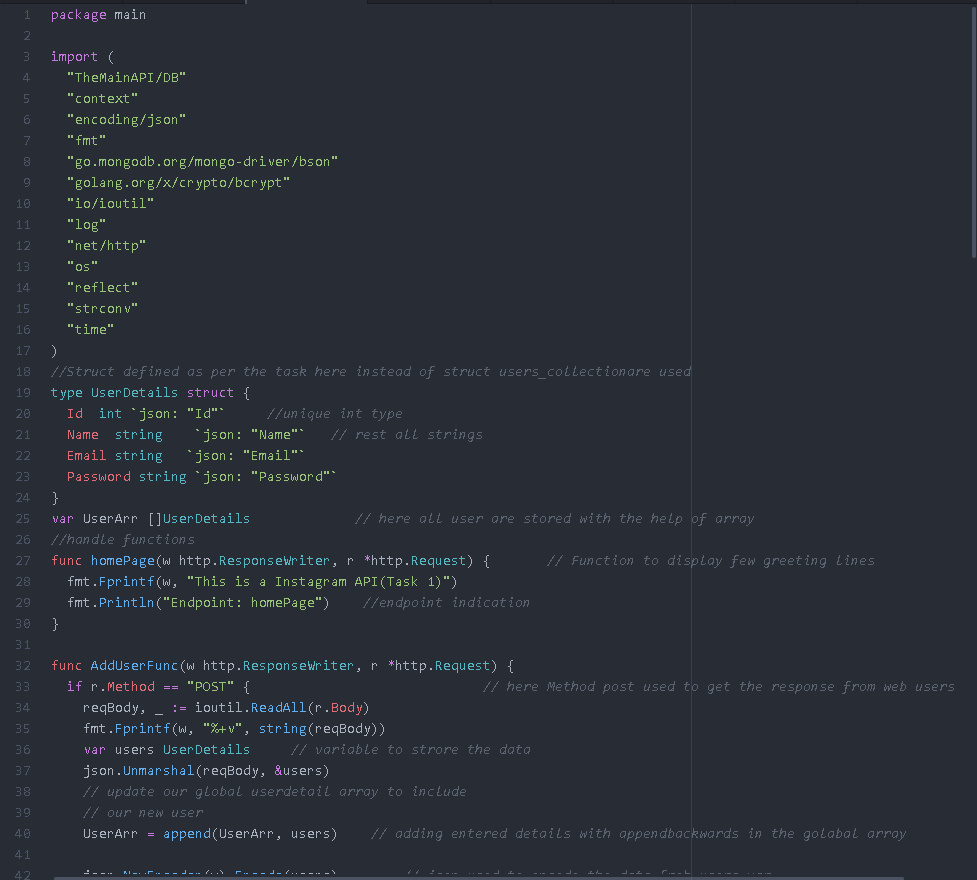
***ThMainAPI folder has been divided as:***

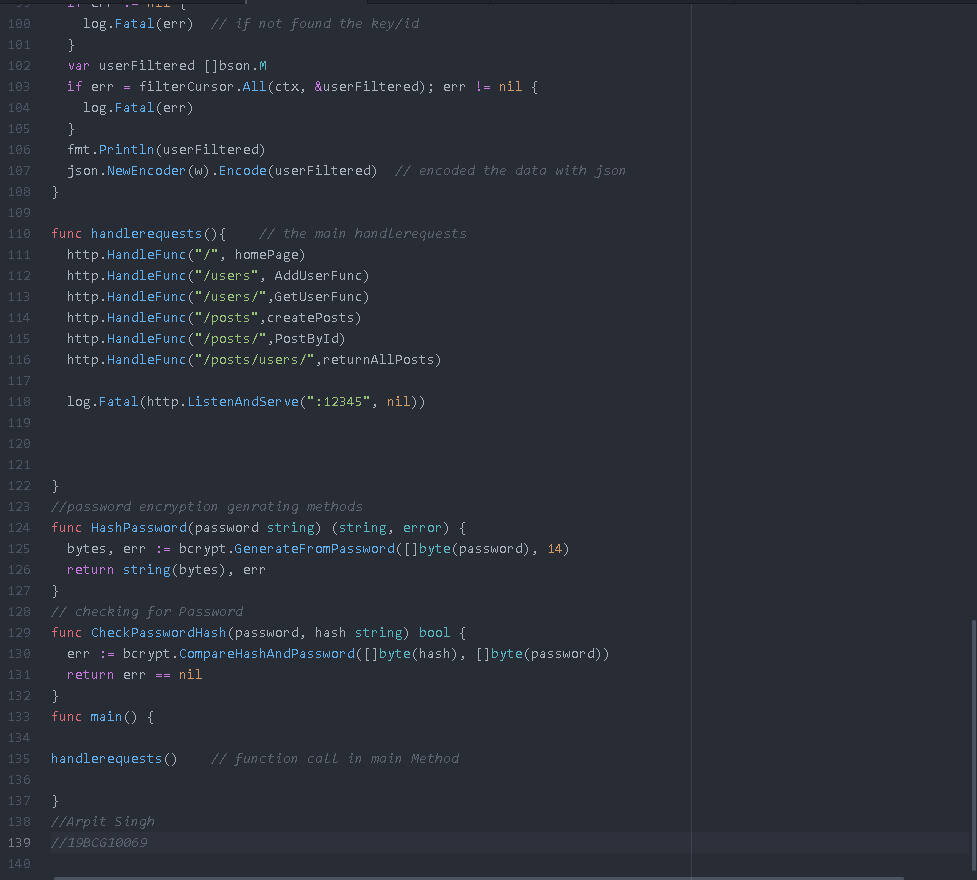
1. ***main.go***
2. ***posts.go***
3. ***go.sum***
4. ***go.mod***
5. ***Database(Folder > db.go)***

***In* Main.go:**

**The basic data model of our application has been implemented with proper http requests and json format.**

**The User data model is implemented and data is stored in the collections accordingly.**

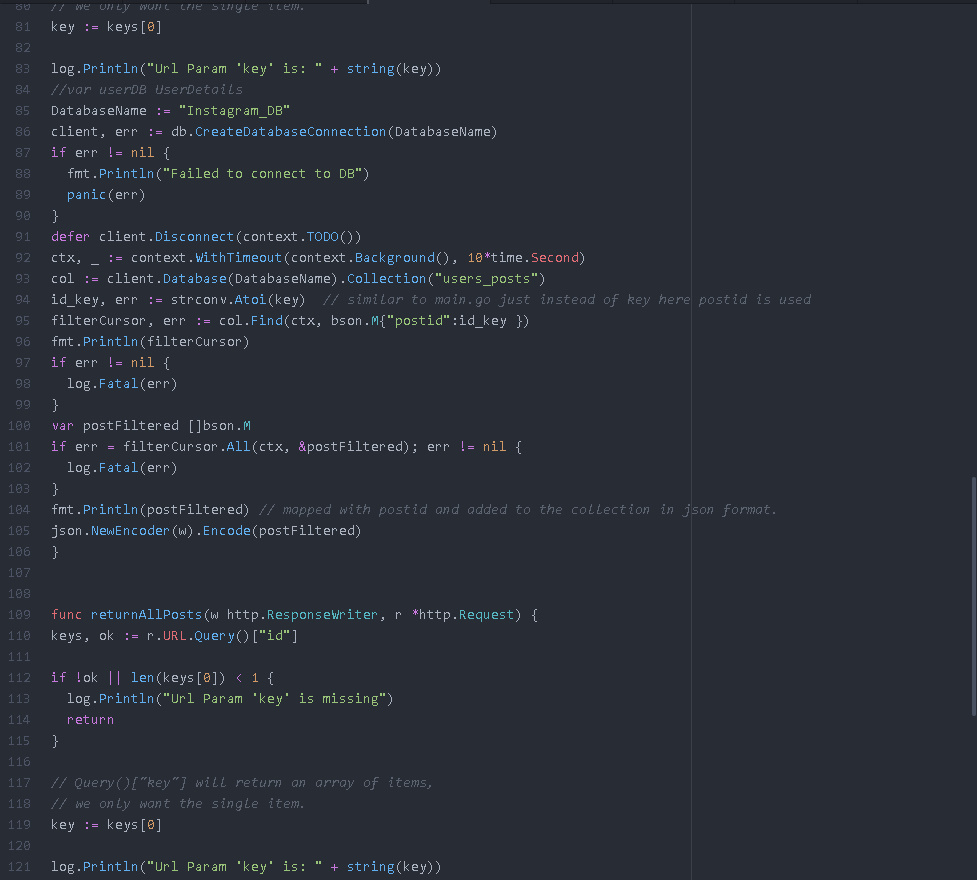
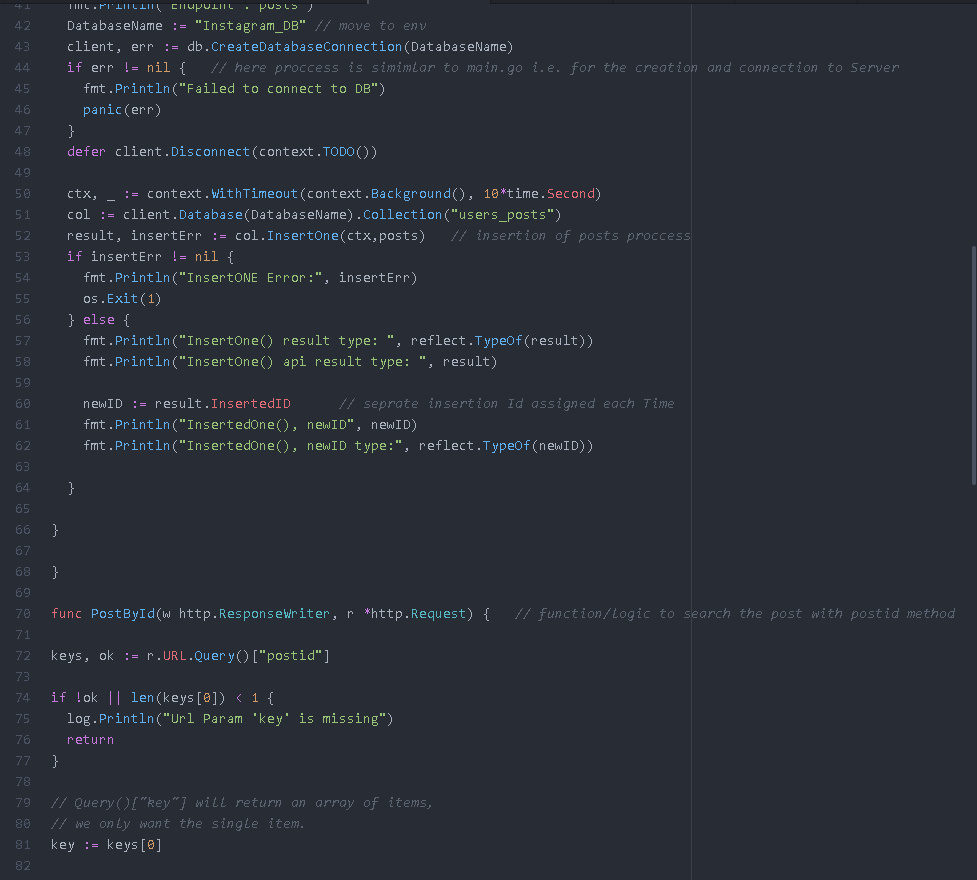
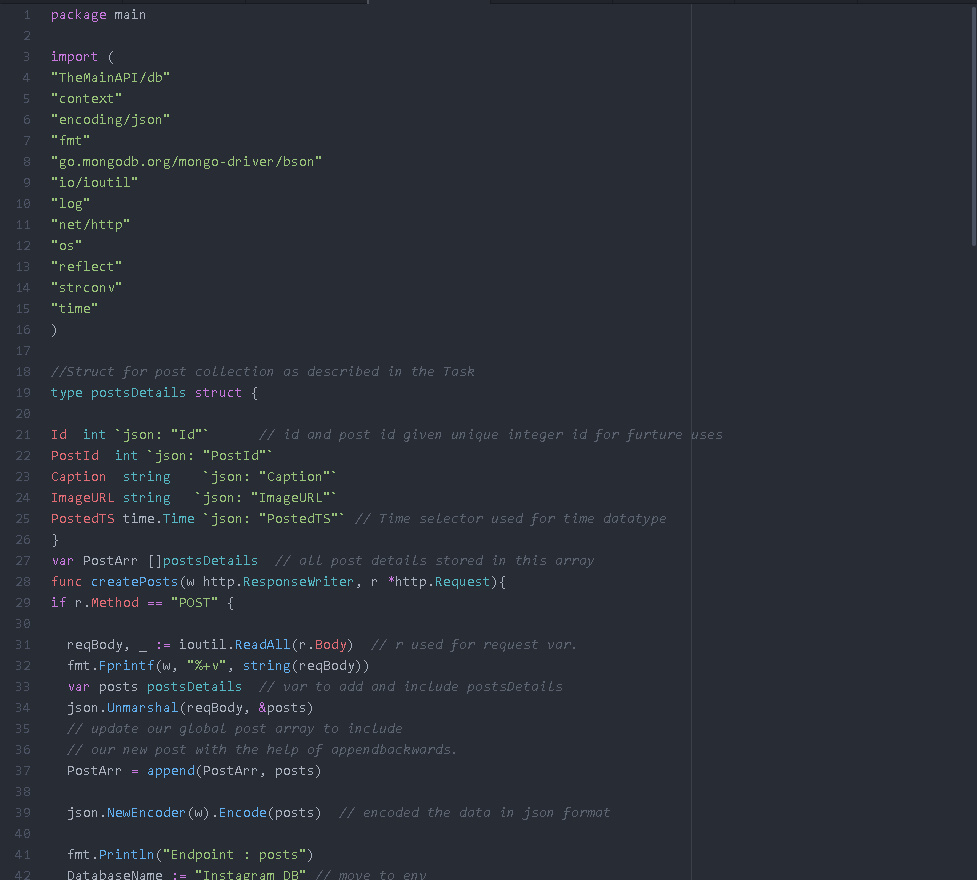
**All the handlers are added and stored in a function named handelfunction().**

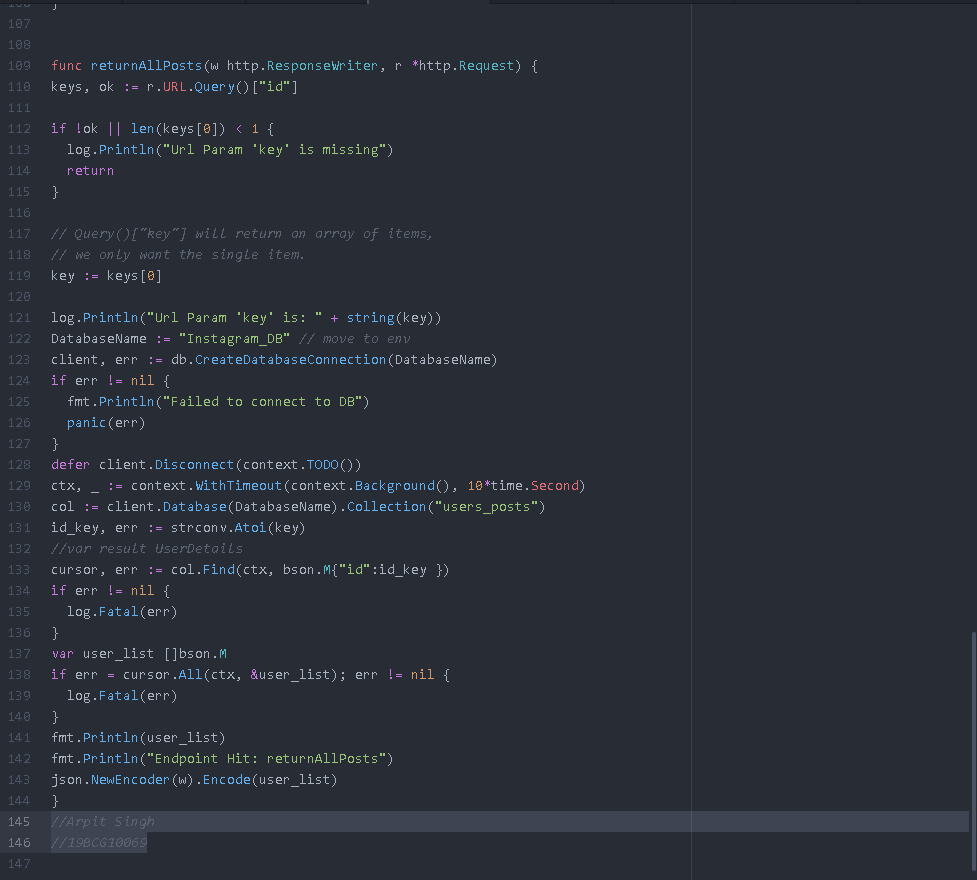


**Next *Posts.go* similar to main.to :**

**However here post data model is described**

**And search by postid/id logic has been implemented.**

**Main.go**

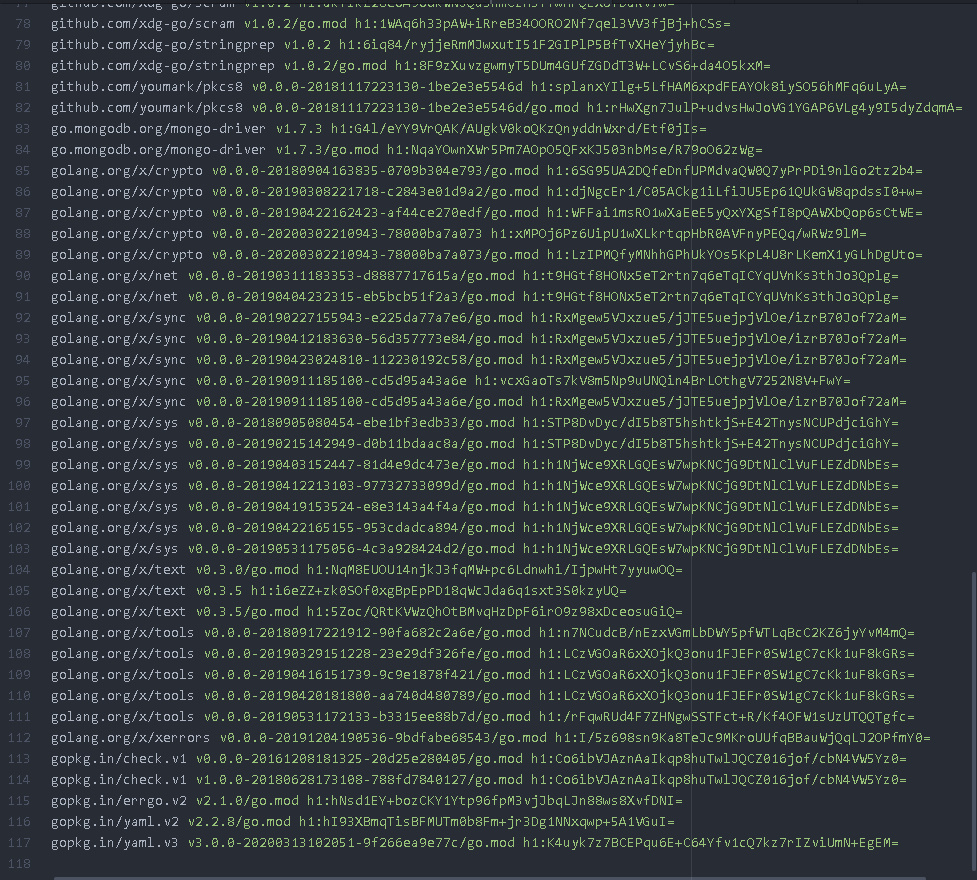
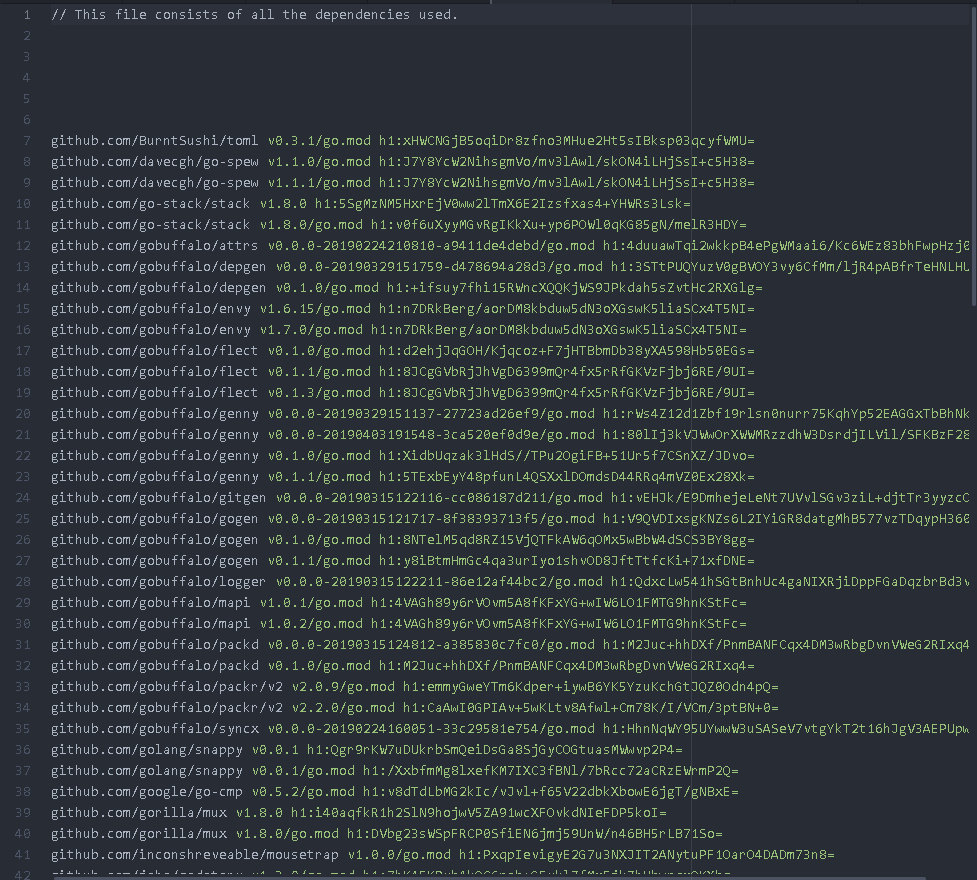


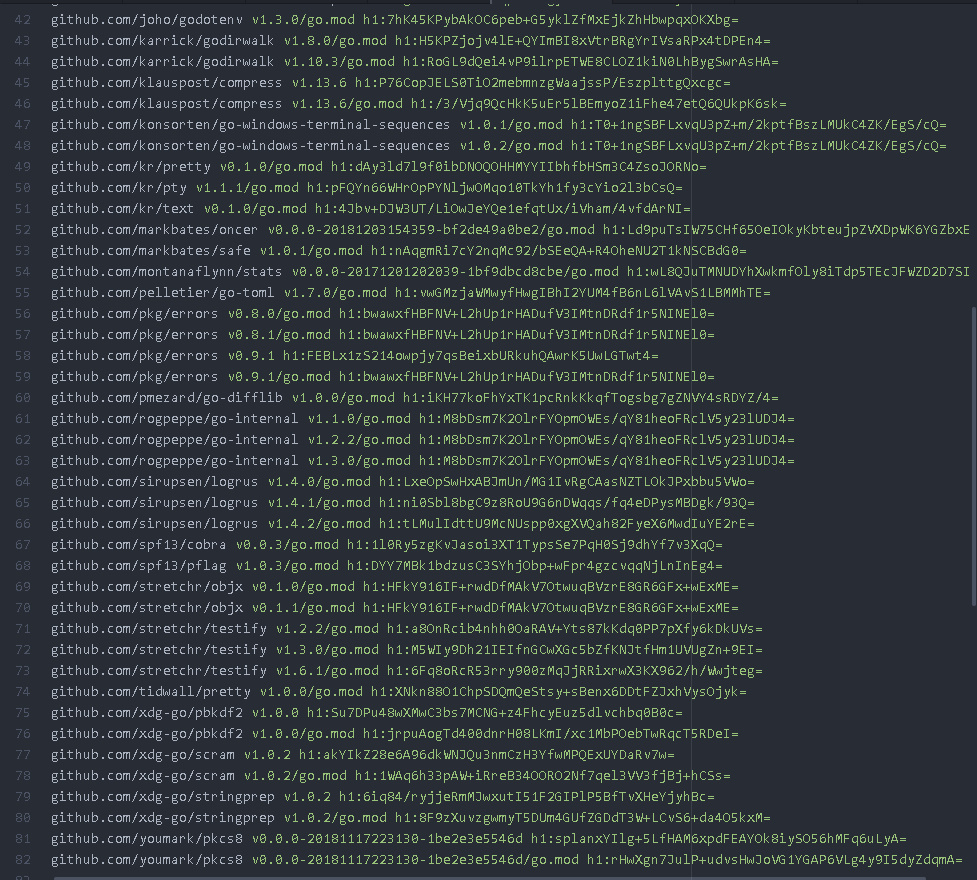
**Next we have *go.sum and go.mod:***

**These 2 files are used to store and get all the required dependencies.**

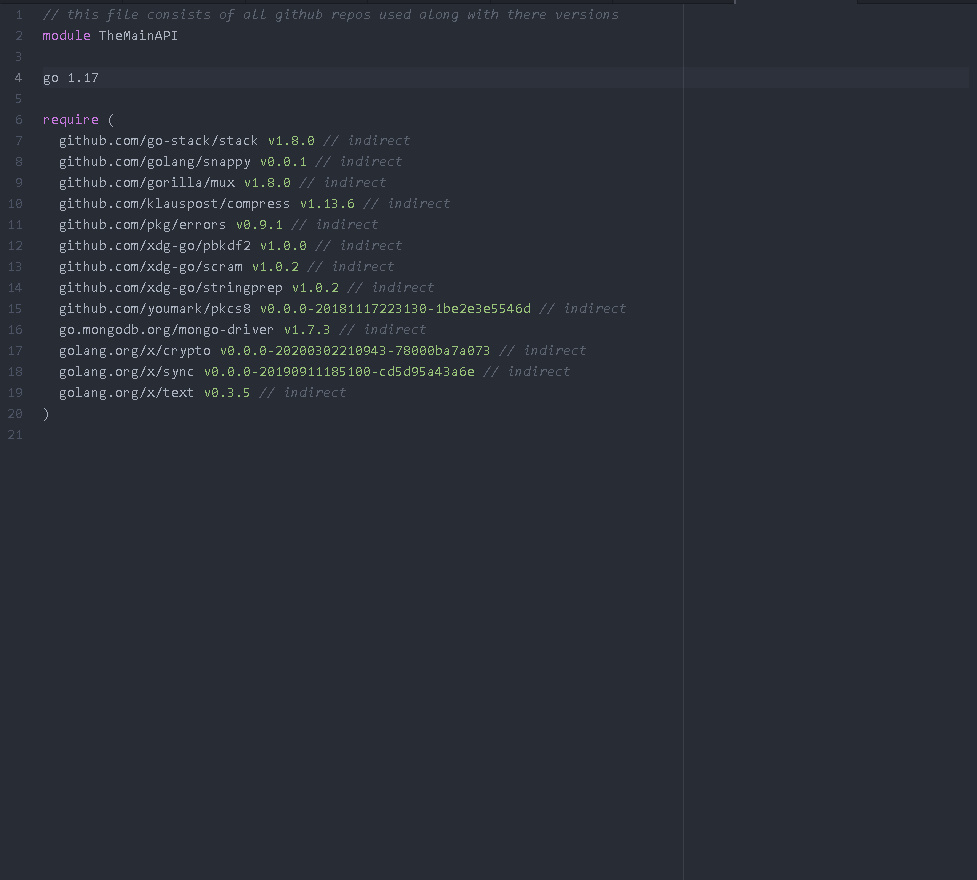
**Mod – module.**

**Sum – checks the cryptographic checksums.**

**GO.SUM: **



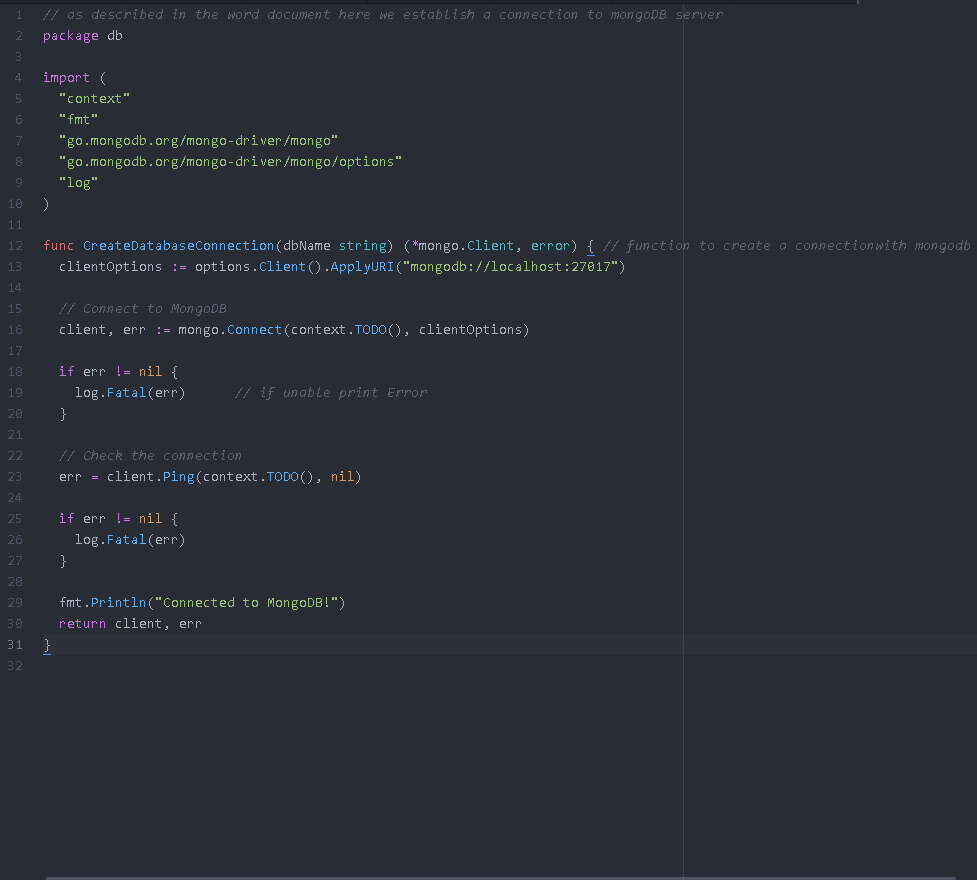
**GO.MOD:**



**Lastly we have a Folder Database(Database>db.go):**

**Here we setup a connection with MongoDB server and Check it as well \*(as done at start of document).**

**Db.go**



**The Folder .Xmlfiles has been added to give browser feasibility.**

**All the files can be accessed at:**

[**https://github.com/TSM-ArpitSG/InstagramAPI**](https://github.com/TSM-ArpitSG/InstagramAPI)