Encoding/json -> used to perform operations related to json.

It performs marshalling-> json.Marshal() and unmarshalling ->json.Unmarshal()

Marshalling -> Converting go objects into json

This method takes object as parameter and returns byte code.

Unmarshalling -> The Method that is used to convert json (byte data) into struct.

Marshalling of struct into json, now we will take json string and unmarshall that json into a struct.

Net/http :-

Package http provides HTTP client and server implementations.

Get, Head, Post and Put make HTTP (or HTTPS) requests.

Gorilla mux:-

Package mux implements a request router and dispatcher.

Mux is HTTP request multiplexer.

Router matches incoming requests against all the registered routes and calls a handler for a route that matches the URL or other conditions.

Gorm:

The GORM is fantastic ORM library for Golang, aims to be developer friendly.

It is an ORM library for dealing with relational databases.

This gorm library is developed on the top of database/sql package.

I used gorm to do the operations on the database.

Db,err := gorm.Open(“mysql”, “user:password@tcp/dbname)

I have connected to the database using DNS link

Defined a user struct by using gorm.Model and also defined fields in json format using encoding/json library.

I initialized migration i.e. initialized database connection.

Defined 5 handler functions to create user, get users, get user, update user, delete user.

Also initialized router using mux.NewRouter did this using gorillamux

The connections are displayed using http here we use net/http.

I performed post,get,put,delete operations in postman.

And saved the data in mysql workbench.