1.Test Program: Creating a go lang rest server application which should connect to Database for storing and retrieving student details.Using Postman for testing API

User.go file

package main

import (

    "encoding/json"

    "fmt"

    "net/http"

    "github.com/gorilla/mux"

    "gorm.io/driver/mysql"

    "gorm.io/gorm"

)

var DB \*gorm.DB

var err error

const DNS = "root:2004@tcp(127.0.0.1:3306)/learning?charset=utf8mb4&parseTime=True&loc=Local"

type User struct {

    gorm.Model

    FirstName string `json:"firstname"`

    LastName  string `json:"lastname"`

    Email     string `json:"email"`

    Mark      string `json:"mark"`

    Class     string `json:"class"`

}

func InitialMigration() {

    DB, err = gorm.Open(mysql.Open(DNS), &gorm.Config{})

    if err != nil {

        fmt.Println(err.Error())

        panic("Cannot connect to DB")

    }

    DB.AutoMigrate(&User{})

}

func GetUsers(w http.ResponseWriter, r \*http.Request) {

    w.Header().Set("Content-Type", "application/json")

    var users1 []User

    DB.Find(&users1)

    json.NewEncoder(w).Encode(users1)

}

func GetUser(w http.ResponseWriter, r \*http.Request) {

    w.Header().Set("Content-Type", "application/json")

    params := mux.Vars(r)

    var user User

    DB.First(&user, params["id"])

    json.NewEncoder(w).Encode(user)

}

func CreateUser(w http.ResponseWriter, r \*http.Request) {

    w.Header().Set("Content-Type", "application/json")

    var user User

    json.NewDecoder(r.Body).Decode(&user)

    DB.Create(&user)

    json.NewEncoder(w).Encode(user)

}

func UpdateUser(w http.ResponseWriter, r \*http.Request) {

    w.Header().Set("Content-Type", "application/json")

    params := mux.Vars(r)

    var user User

    DB.First(&user, params["id"])

    json.NewDecoder(r.Body).Decode(&user)

    DB.Save(&user)

    json.NewEncoder(w).Encode(user)

}

func DeleteUser(w http.ResponseWriter, r \*http.Request) {

    w.Header().Set("Content-Type", "application/json")

    params := mux.Vars(r)

    var user User

    DB.Delete(&user, params["id"])

    json.NewEncoder(w).Encode("The USer is Deleted Successfully!")

}

**Main.go file:**

package main

import (

    "log"

    "net/http"

    "github.com/gorilla/mux"

)

func initializeRouter() {

    r := mux.NewRouter()

    r.HandleFunc("/users", GetUsers).Methods("GET")

    r.HandleFunc("/users/{id}", GetUser).Methods("GET")

    r.HandleFunc("/users", CreateUser).Methods("POST")

    r.HandleFunc("/users/{id}", UpdateUser).Methods("PUT")

    r.HandleFunc("/users/{id}", DeleteUser).Methods("DELETE")

    log.Fatal(http.ListenAndServe(":9000", r))

}

func main() {

    InitialMigration()

    initializeRouter()

}