13th Week

Write a go program for CRUD using MYSQL from scratch

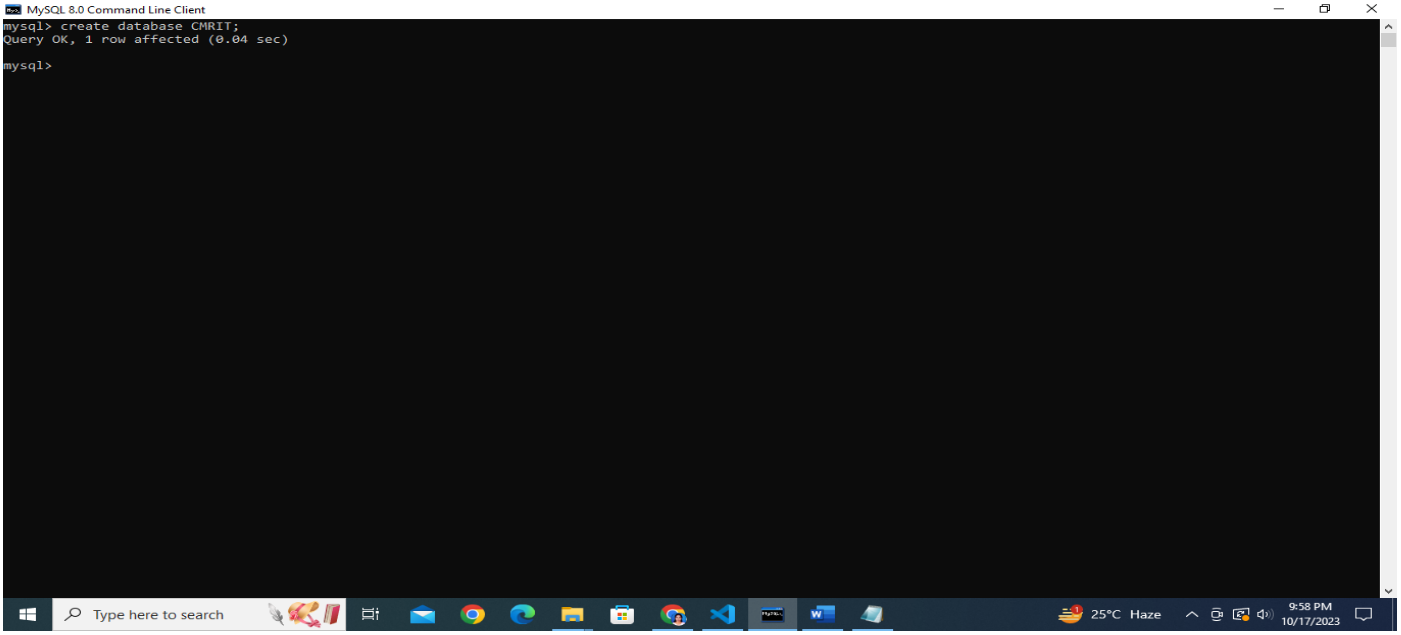
Install MYSQL database with username as “**root**” and password is as ur wish here, I have given my password as “**Yamuna@123**”

Import MYSQL Drivers into our project by using below command in Command Prompt

C:\Users\DELL\Desktop\YamunaDB>**go get -u github.com/go-sql-driver/mysql** .

Now create Database named **CMRIT** by using below command in MYSQL

Mysql> create database CMRIT;

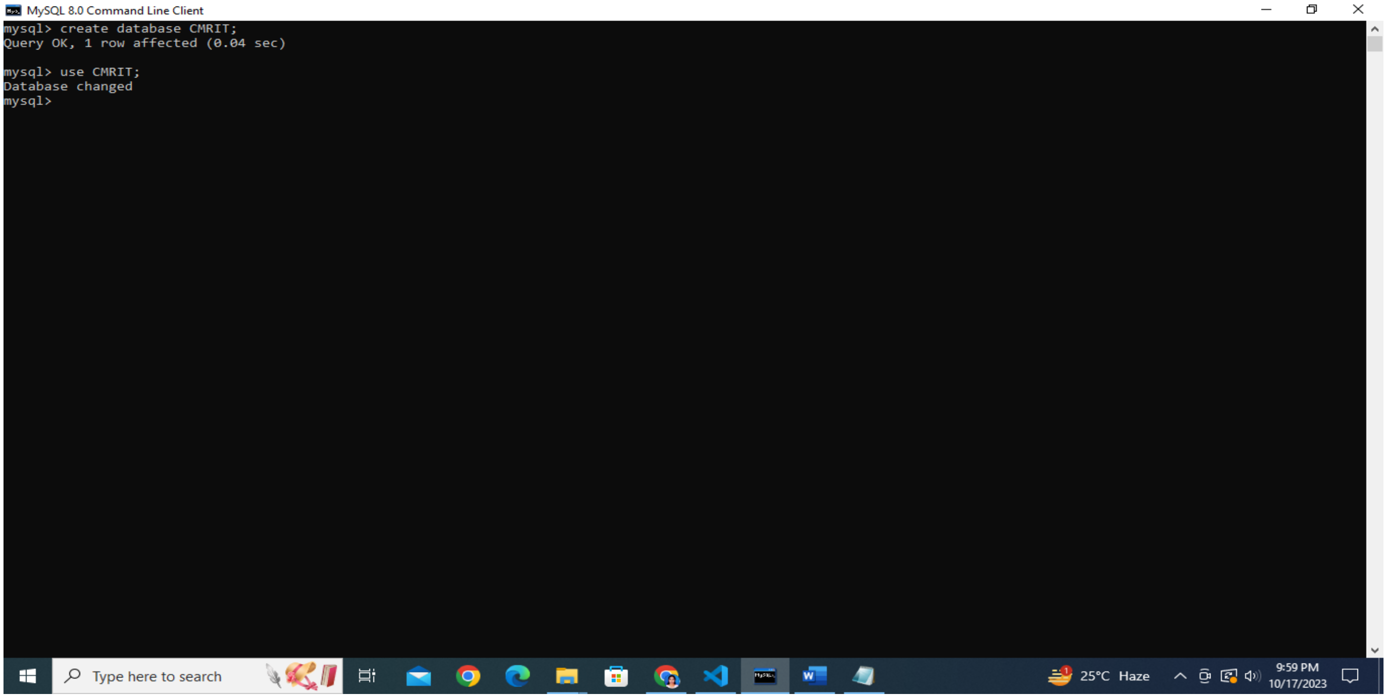


```
mysql> create database CMRIT;
Query OK, 1 row affected (0.04 sec)

mysql>
```

Now enter into that database.

Mysql> **use CMRIT;** //Now we are in CMRIT database



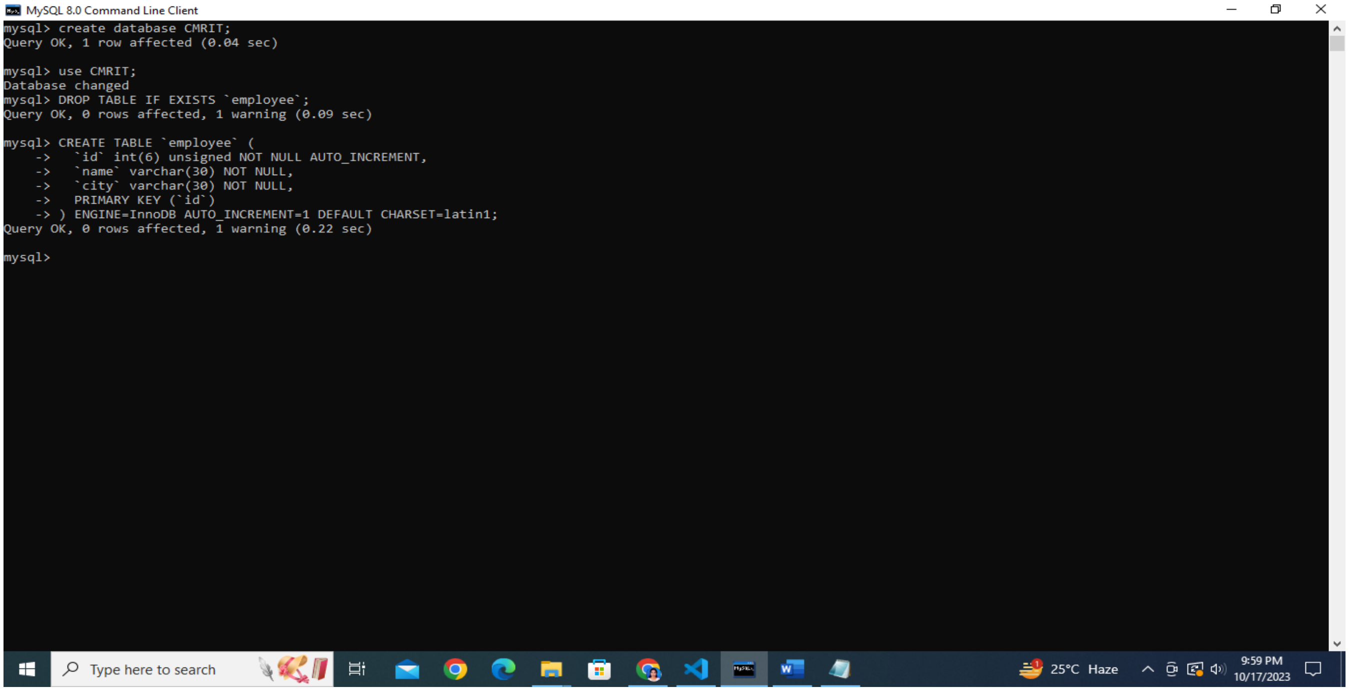
```
mysql> create database CMRIT;
Query OK, 1 row affected (0.04 sec)

mysql> use CMRIT;
Database changed
mysql>
```

Create table Employee by using below command

Mysql> **DROP TABLE IF EXISTS `employee`;**

```
CREATE TABLE `employee` (
  `id` int(6) unsigned NOT NULL AUTO_INCREMENT,
  `name` varchar(30) NOT NULL,
  `city` varchar(30) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;
```

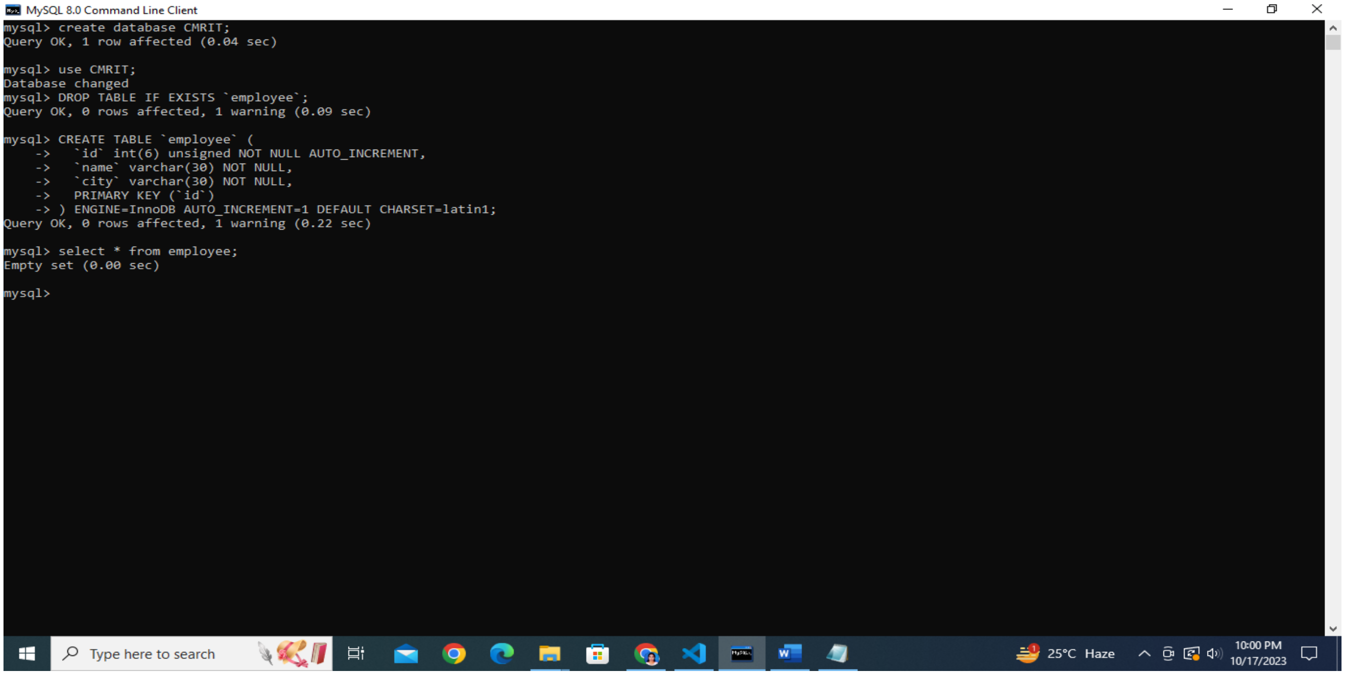


```
mysql> create database CMRIT;
Query OK, 1 row affected (0.04 sec)

mysql> use CMRIT;
Database changed
mysql> DROP TABLE IF EXISTS `employee`;
Query OK, 0 rows affected, 1 warning (0.09 sec)

mysql> CREATE TABLE `employee` (
    ->   `id` int(6) unsigned NOT NULL AUTO_INCREMENT,
    ->   `name` varchar(30) NOT NULL,
    ->   `city` varchar(30) NOT NULL,
    ->   PRIMARY KEY (`id`)
    -> ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;
Query OK, 0 rows affected, 1 warning (0.22 sec)

mysql>
```



```
mysql> create database CMRIT;
Query OK, 1 row affected (0.04 sec)

mysql> use CMRIT;
Database changed
mysql> DROP TABLE IF EXISTS `employee`;
Query OK, 0 rows affected, 1 warning (0.09 sec)

mysql> CREATE TABLE `employee` (
    ->   `id` int(6) unsigned NOT NULL AUTO_INCREMENT,
    ->   `name` varchar(30) NOT NULL,
    ->   `city` varchar(30) NOT NULL,
    ->   PRIMARY KEY (`id`)
    -> ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;
Query OK, 0 rows affected, 1 warning (0.22 sec)

mysql> select * from employee;
Empty set (0.00 sec)

mysql>
```

Create a folder named **YamunaDB**

Let's create a file named main.go and put the following code inside it.

Create Employee struct that has following properties: Id, Name and City.

```
package main

import (
    "database/sql"
    "log"
    "net/http"
    "text/template"
    _ "github.com/go-sql-driver/mysql"
)

type Employee struct {
    Id    int
    Name  string
    City string
}

func dbConn() (db *sql.DB) {
    dbDriver := "mysql"
    dbUser := "root"
    dbPass := "Yamuna@123"
    dbName := "CMRIT"
    db, err := sql.Open(dbDriver, dbUser+":"+dbPass+"@/"+dbName)
    if err != nil {
        panic(err.Error())
    }
    return db
}

var tmpl = template.Must(template.ParseGlob("form/*"))

func Index(w http.ResponseWriter, r *http.Request) {
    db := dbConn()
    selDB, err := db.Query("SELECT * FROM Employee ORDER BY id DESC")
    if err != nil {
        panic(err.Error())
    }
    emp := Employee{}
    res := []Employee{}
    for selDB.Next() {
        var id int
        var name, city string
        err = selDB.Scan(&id, &name, &city)
        if err != nil {
            panic(err.Error())
        }
        emp.Id = id
        emp.Name = name
        emp.City = city
        res = append(res, emp)
    }
    tmpl.ExecuteTemplate(w, "Index", res)
    defer db.Close()
}

func Show(w http.ResponseWriter, r *http.Request) {
    db := dbConn()
    nId := r.URL.Query().Get("id")
    selDB, err := db.Query("SELECT * FROM Employee WHERE id=?", nId)
    if err != nil {
        panic(err.Error())
    }
    emp := Employee{}
    for selDB.Next() {
        var id int
        var name, city string
        err = selDB.Scan(&id, &name, &city)
        if err != nil {
            panic(err.Error())
        }
        emp.Id = id
        emp.Name = name
        emp.City = city
    }
    tmpl.ExecuteTemplate(w, "Show", emp)
    defer db.Close()
}
func New(w http.ResponseWriter, r *http.Request) {
    tmpl.ExecuteTemplate(w, "New", nil)
}
func Edit(w http.ResponseWriter, r *http.Request) {
    db := dbConn()
    nId := r.URL.Query().Get("id")
    selDB, err := db.Query("SELECT * FROM Employee WHERE id=?", nId)
    if err != nil {
        panic(err.Error())
    }
    emp := Employee{}
    for selDB.Next() {
        var id int
        var name, city string
        err = selDB.Scan(&id, &name, &city)
        if err != nil {
            panic(err.Error())
        }
        emp.Id = id
        emp.Name = name
        emp.City = city
    }
    tmpl.ExecuteTemplate(w, "Edit", emp)
    defer db.Close()
}
```

```
func Insert(w http.ResponseWriter, r *http.Request) {
    db := dbConn()
    if r.Method == "POST" {
        name := r.FormValue("name")
        city := r.FormValue("city")
        insForm, err := db.Prepare("INSERT INTO Employee(name, city) VALUES(?,?)")
        if err != nil {
            panic(err.Error())
        }
        insForm.Exec(name, city)
        log.Println("INSERT: Name: " + name + " | City: " + city)
    }
    defer db.Close()
    http.Redirect(w, r, "/", 301)
}
func Update(w http.ResponseWriter, r *http.Request) {
    db := dbConn()
    if r.Method == "POST" {
        name := r.FormValue("name")
        city := r.FormValue("city")
        id := r.FormValue("uid")
        insForm, err := db.Prepare("UPDATE Employee SET name=?, city=? WHERE id=?")
        if err != nil {
            panic(err.Error())
        }
        insForm.Exec(name, city, id)
        log.Println("UPDATE: Name: " + name + " | City: " + city)
    }
    defer db.Close()
    http.Redirect(w, r, "/", 301)
}

func Delete(w http.ResponseWriter, r *http.Request) {
    db := dbConn()
    emp := r.URL.Query().Get("id")
    delForm, err := db.Prepare("DELETE FROM Employee WHERE id=?")
    if err != nil {
        panic(err.Error())
    }
    delForm.Exec(emp)
    log.Println("DELETE")
    defer db.Close()
    http.Redirect(w, r, "/", 301)
}
func main() {
    log.Println("Server started on: http://localhost:8080")
    http.HandleFunc("/", Index)
    http.HandleFunc("/show", Show)
    http.HandleFunc("/new", New)
    http.HandleFunc("/edit", Edit)
    http.HandleFunc("/insert", Insert)
    http.HandleFunc("/update", Update)
    http.HandleFunc("/delete", Delete)
    http.ListenAndServe(":8080", nil)
}
```

Creating Template Files

Template files of our CRUD (Create, read, Update, Delete) application

Create **form** folder at same location where we have created **main.go**

…….Create a file named Index.tmpl inside the form folder

…….Create a file named Header.tmpl inside the form folder

…….Create a file named Footer.tmpl inside the form folder

…….Create a file named Menu.tmpl inside the form folder

…….Create a file named Show.tmpl inside the form folder

…….Create a file named New.tmpl inside the form folder

…….Create a file named Edit.tmpl inside the form folder

1. Create a file named Index.tmpl

```
{{ define "Index" }}
  {{ template "Header" }}
    {{ template "Menu"  }}
    <h2> Registered </h2>
    <table border="1">
      <thead>
      <tr>
        <td>ID</td>
        <td>Name</td>
        <td>City</td>
        <td>View</td>
        <td>Edit</td>
        <td>Delete</td>
      </tr>
       </thead>
       <tbody>
    {{ range . }}
      <tr>
        <td>{{ .Id }}</td>
        <td> {{ .Name }} </td>
        <td>{{ .City }} </td>
        <td><a href="/show?id={{ .Id }}">View</a></td>
        <td><a href="/edit?id={{ .Id }}">Edit</a></td>
        <td><a href="/delete?id={{ .Id }}">Delete</a><td>
      </tr>
    {{ end }}
       </tbody>
    </table>
  {{ template "Footer" }}
{{ end }}
```

2. **Create a file named Header.tmpl inside the form folder**

```
{{ define "Header" }}
<!DOCTYPE html>
<html lang="en-US">
    <head>
        <title>Golang Mysql Curd Example</title>
        <meta charset="UTF-8" />
    </head>
    <body>
        <h1>Golang Mysql Curd Example</h1>
{{ end }}
```

3. **Create a file named Footer.tmpl inside the form folder**

```
{{ define "Footer" }}
    </body>
</html>
{{ end }}
```

4. **Create a file named Menu.tmpl inside the form folder**

```
{{ define "Menu" }}
<a href="/">HOME</a> |
<a href="/new">NEW</a>
{{ end }}
```

5. **Create a file named Show.tmpl inside the form folder**

```
{{ define "Show" }}
  {{ template "Header" }}
    {{ template "Menu"  }}
    <h2> Register {{ .Id }} </h2>
      <p>Name: {{ .Name }}</p>
      <p>City:  {{ .City }}</p><br /> <a href="/edit?id={{ .Id }}">Edit</a></p>
  {{ template "Footer" }}
{{ end }}
```

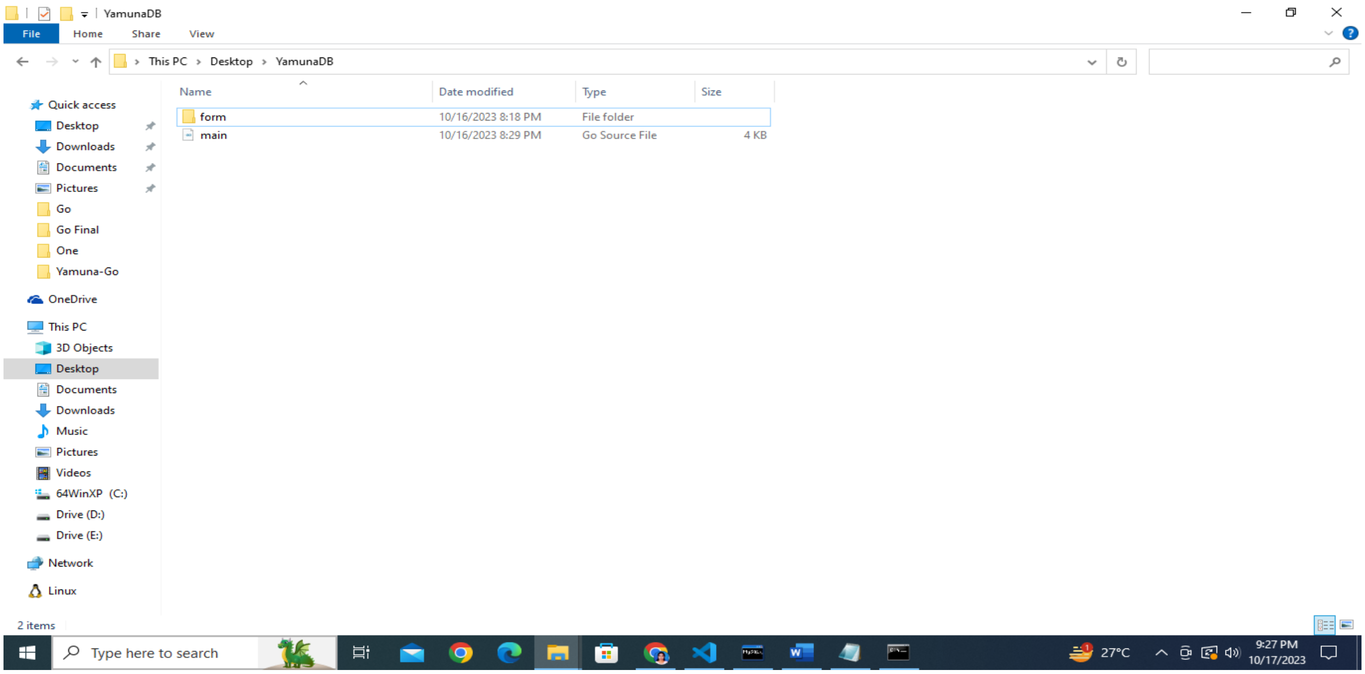
6. **Create a file named New.tmpl inside the form folder**

```
{{ define "New" }}
  {{ template "Header" }}
    {{ template "Menu" }}
   <h2>New Name and City</h2>
```

```
    <form method="POST" action="insert">
      <label> Name </label><input type="text" name="name" /><br />
      <label> City </label><input type="text" name="city" /><br />
      <input type="submit" value="Save user" />
    </form>
  {{ template "Footer" }}
{{ end }}
```

7. Create a file named Edit.tmpl inside the form folder

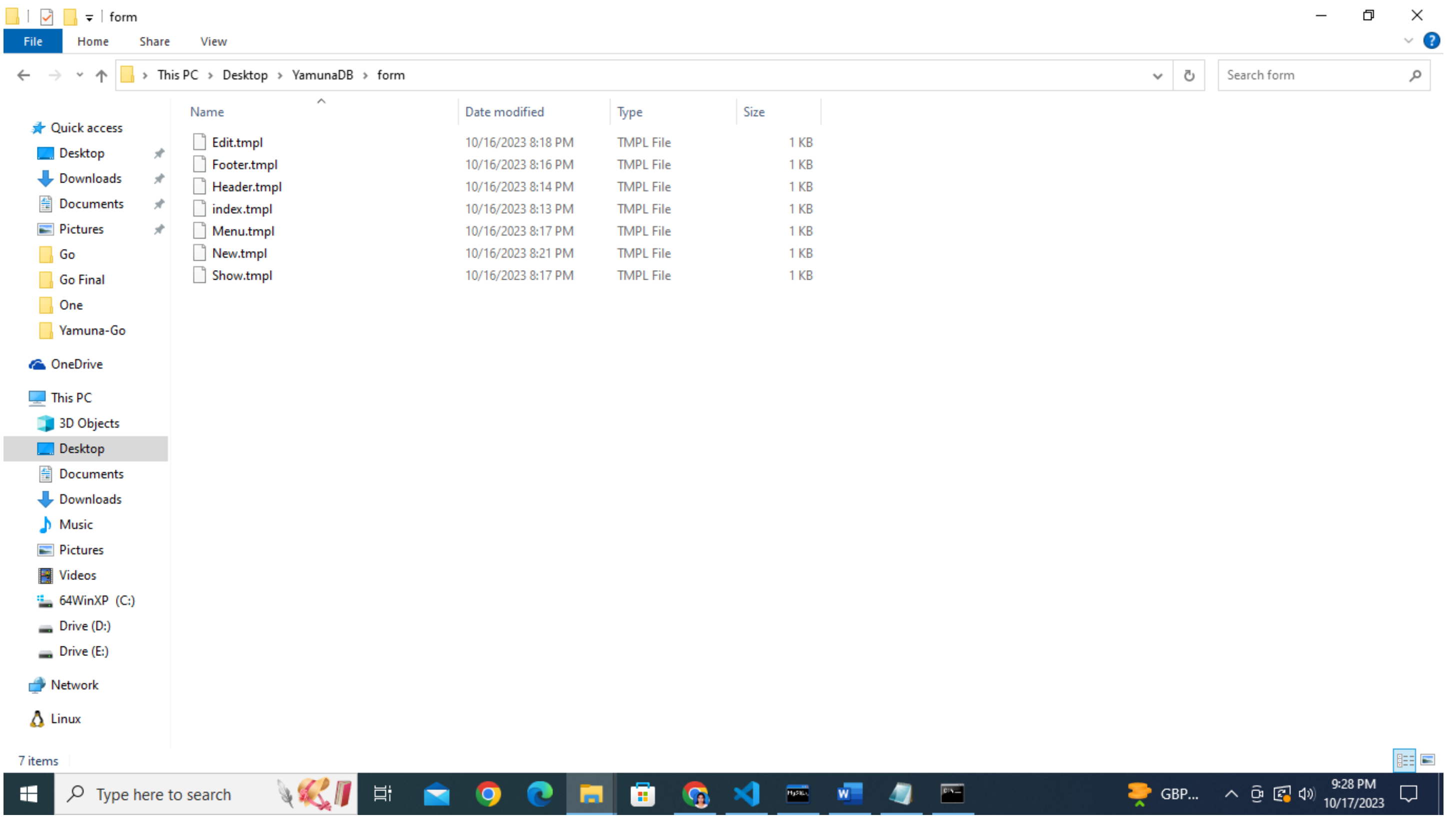
```
{{ define "Edit" }}
  {{ template "Header" }}
    {{ template "Menu" }}
   <h2>Edit Name and City</h2>
    <form method="POST" action="update">
      <input type="hidden" name="uid" value="{{ .Id }}" />
      <label> Name </label><input type="text" name="name" value="{{ .Name }}"  /><br />
      <label> City </label><input type="text" name="city" value="{{ .City }}"  /><br />
      <input type="submit" value="Save user" />
    </form><br />
  {{ template "Footer" }}
{{ end }}
```



This PC > Desktop > YamunaDB

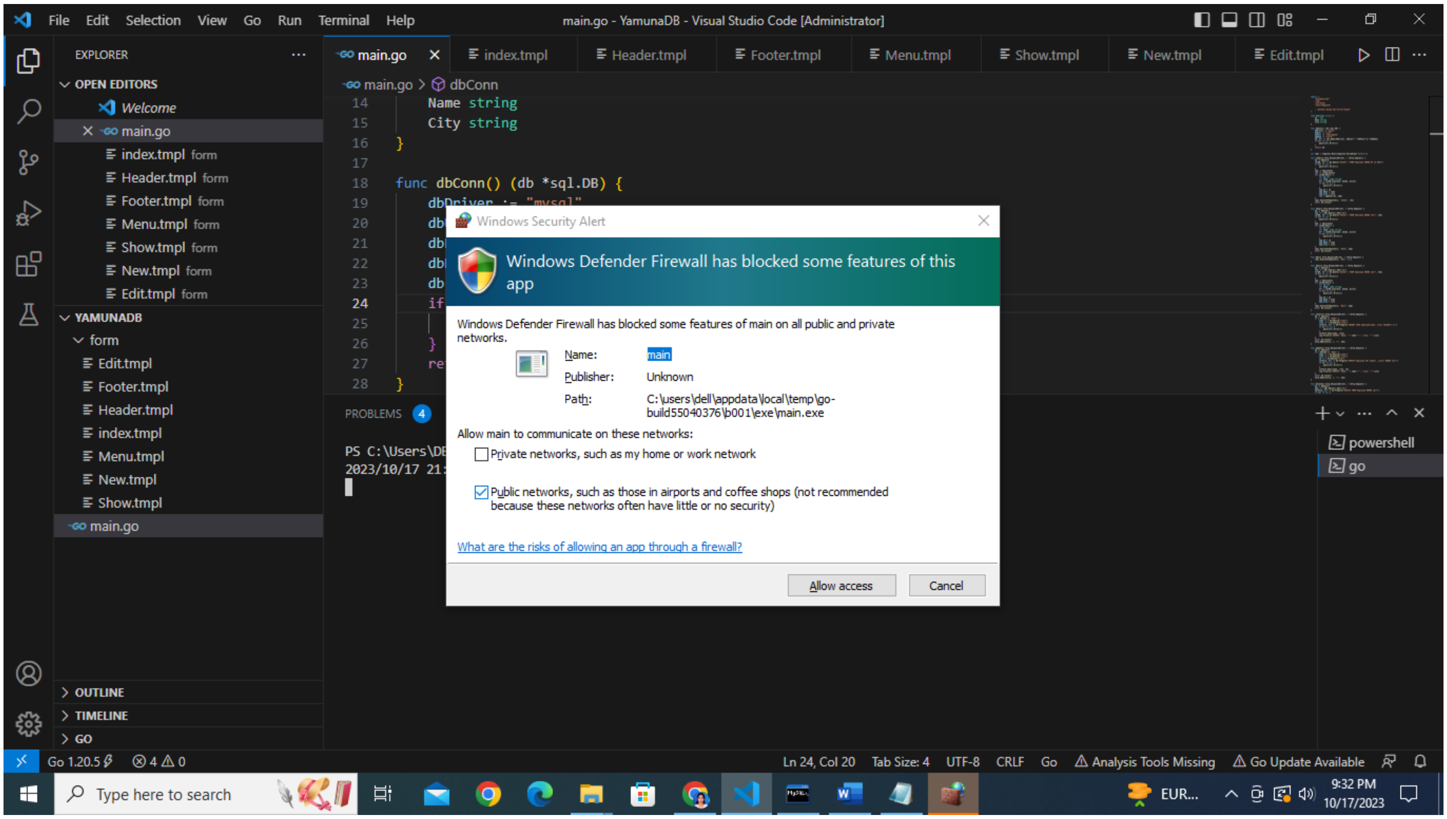
Name	Date modified	Type	Size
form	10/16/2023 8:18 PM	File folder	
main	10/16/2023 8:29 PM	Go Source File	4 KB

2 items



Name	Date modified	Type	Size
Edit.tmpl	10/16/2023 8:18 PM	TMPL File	1 KB
Footer.tmpl	10/16/2023 8:16 PM	TMPL File	1 KB
Header.tmpl	10/16/2023 8:14 PM	TMPL File	1 KB
index.tmpl	10/16/2023 8:13 PM	TMPL File	1 KB
Menu.tmpl	10/16/2023 8:17 PM	TMPL File	1 KB
New.tmpl	10/16/2023 8:21 PM	TMPL File	1 KB
Show.tmpl	10/16/2023 8:17 PM	TMPL File	1 KB

Run the following command



```
    Name string
    City string
}

func dbConn() (db *sql.DB) {
    dbDriver := "mysql"
    db
    db
    db
    db
    if
    |
    }
    re
}
```

```
PS C:\Users\DE
2023/10/17 21:
```

Windows Security Alert

Windows Defender Firewall has blocked some features of this app

Windows Defender Firewall has blocked some features of main on all public and private networks.

Name: main

Publisher: Unknown

Path: C:\users\dell\appdata\local\temp\go-build55040376\b001\exe\main.exe

Allow main to communicate on these networks:

☐ Private networks, such as my home or work network

☑ Public networks, such as those in airports and coffee shops (not recommended because these networks often have little or no security)

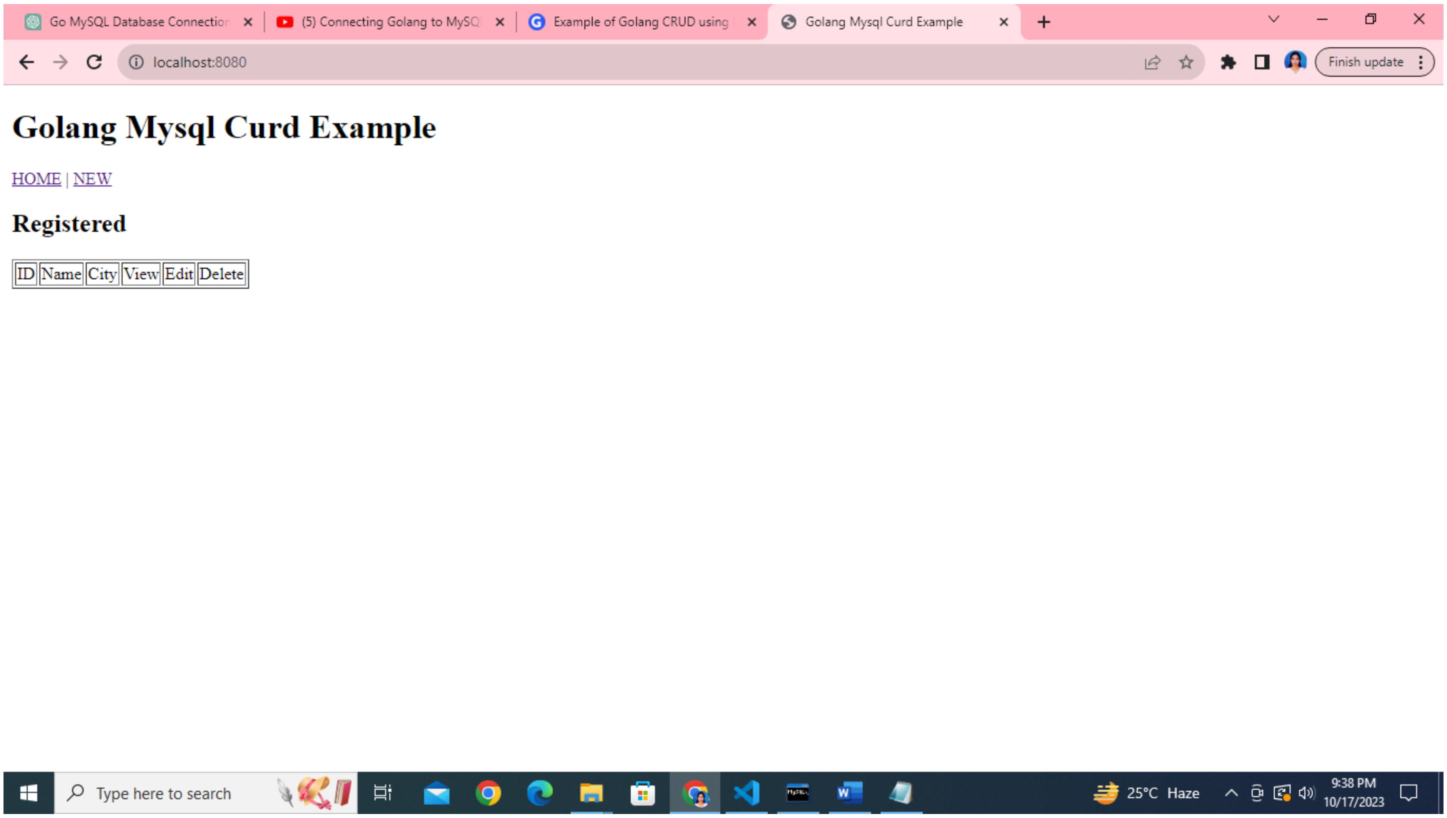
What are the risks of allowing an app through a firewall?

Allow access | Cancel

Click on Allow access, and go to Crome

Load the following URL as http://localhost:8080/
Now we will see the output as below

Click on **New**

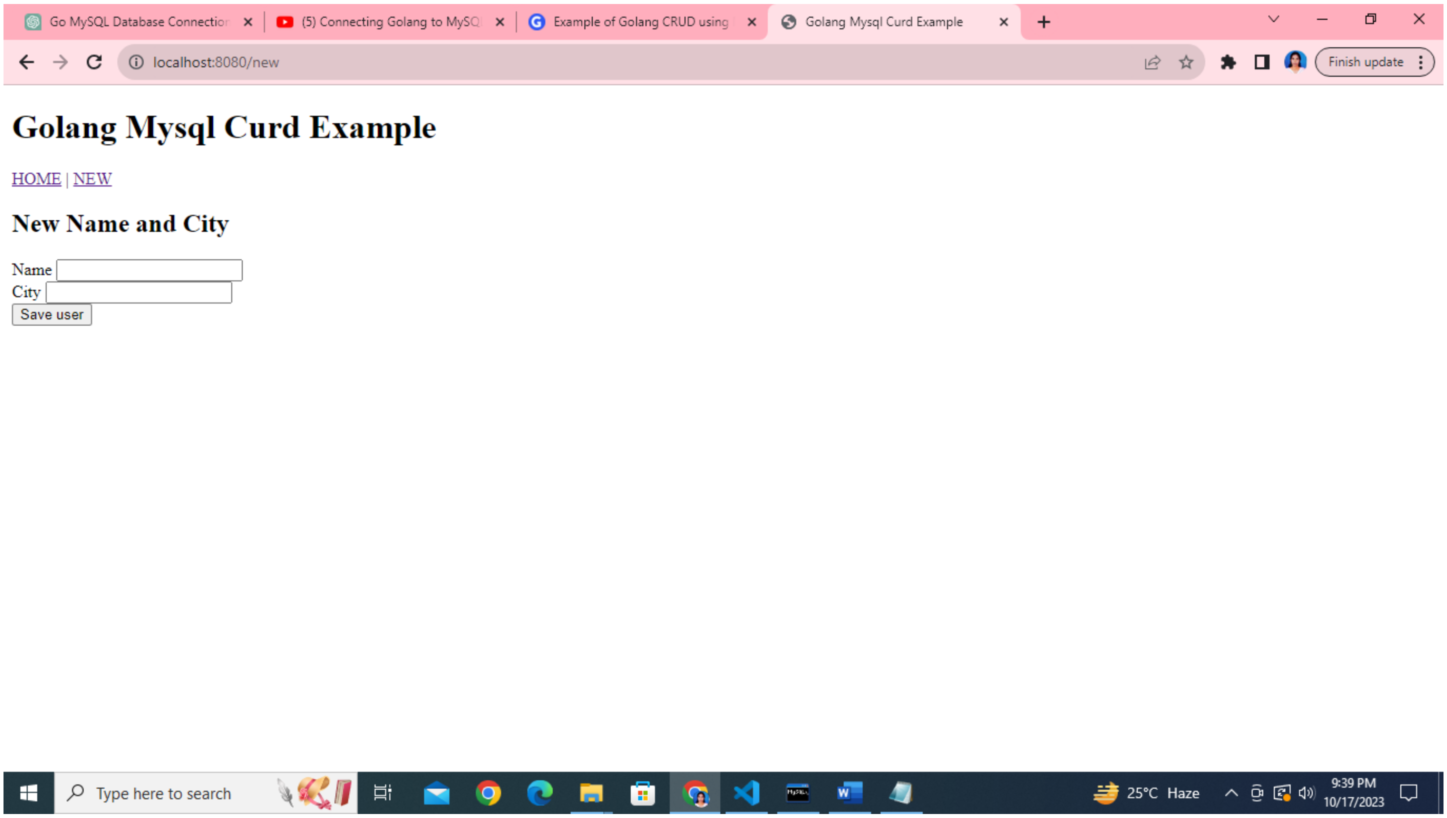

Go MySQL Database Connectio
(5) Connecting Golang to MySQ
Example of Golang CRUD using
Golang Mysql Curd Example
localhost:8080
Finish update
Type here to search
25°C Haze
9:38 PM
10/17/2023

Golang Mysql Curd Example

HOME | NEW

Registered

ID	Name	City	View	Edit	Delete


Go MySQL Database Connectio
(5) Connecting Golang to MySQ
Example of Golang CRUD using
Golang Mysql Curd Example
localhost:8080/new
Finish update
Type here to search
25°C Haze
9:39 PM
10/17/2023

Golang Mysql Curd Example

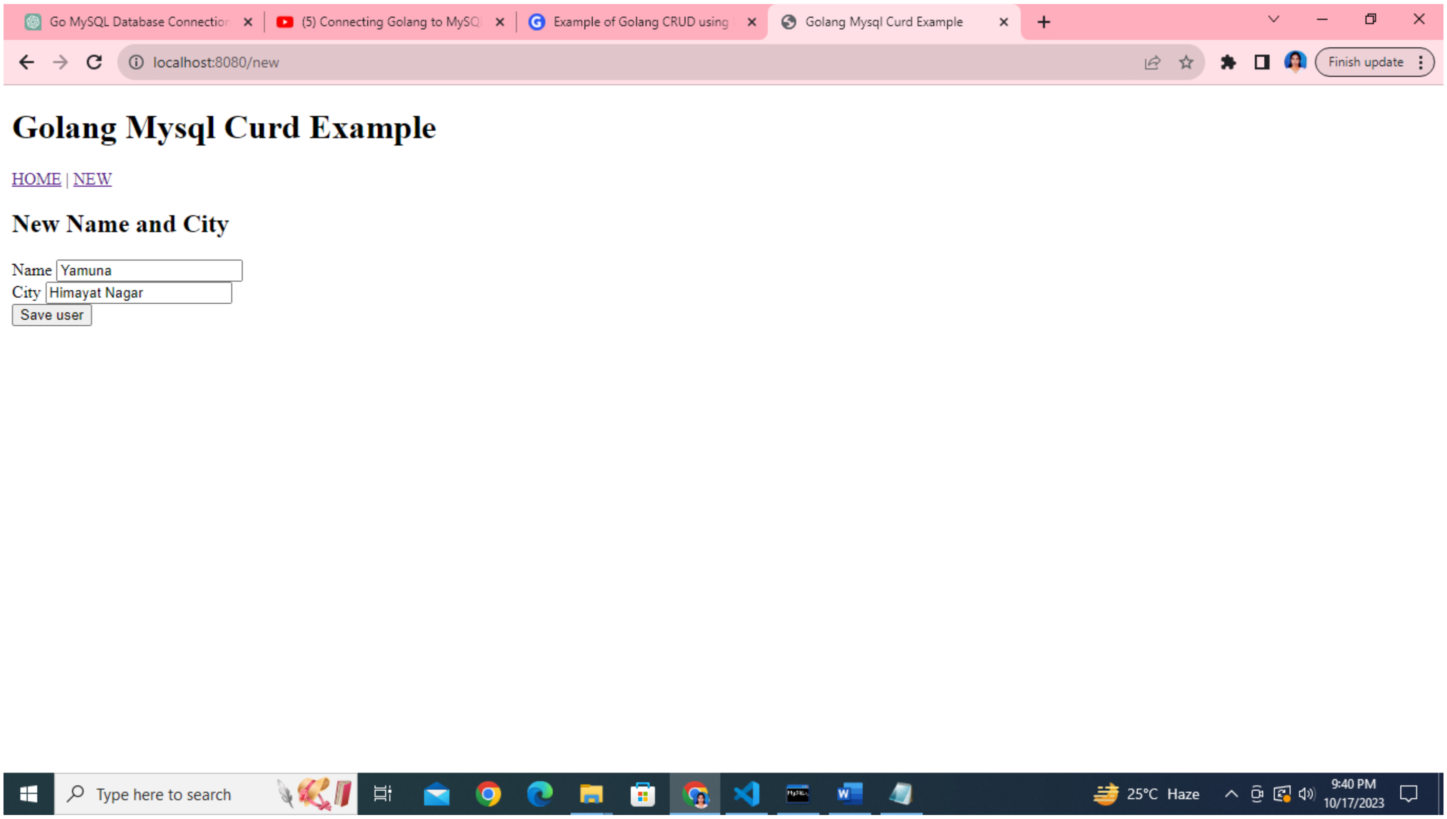
HOME | NEW

New Name and City

Name

City

Save user


Go MySQL Database Connectior
(5) Connecting Golang to MySQ
Example of Golang CRUD using
Golang Mysql Curd Example
localhost:8080/new
Finish update
Type here to search
25°C Haze
9:40 PM
10/17/2023

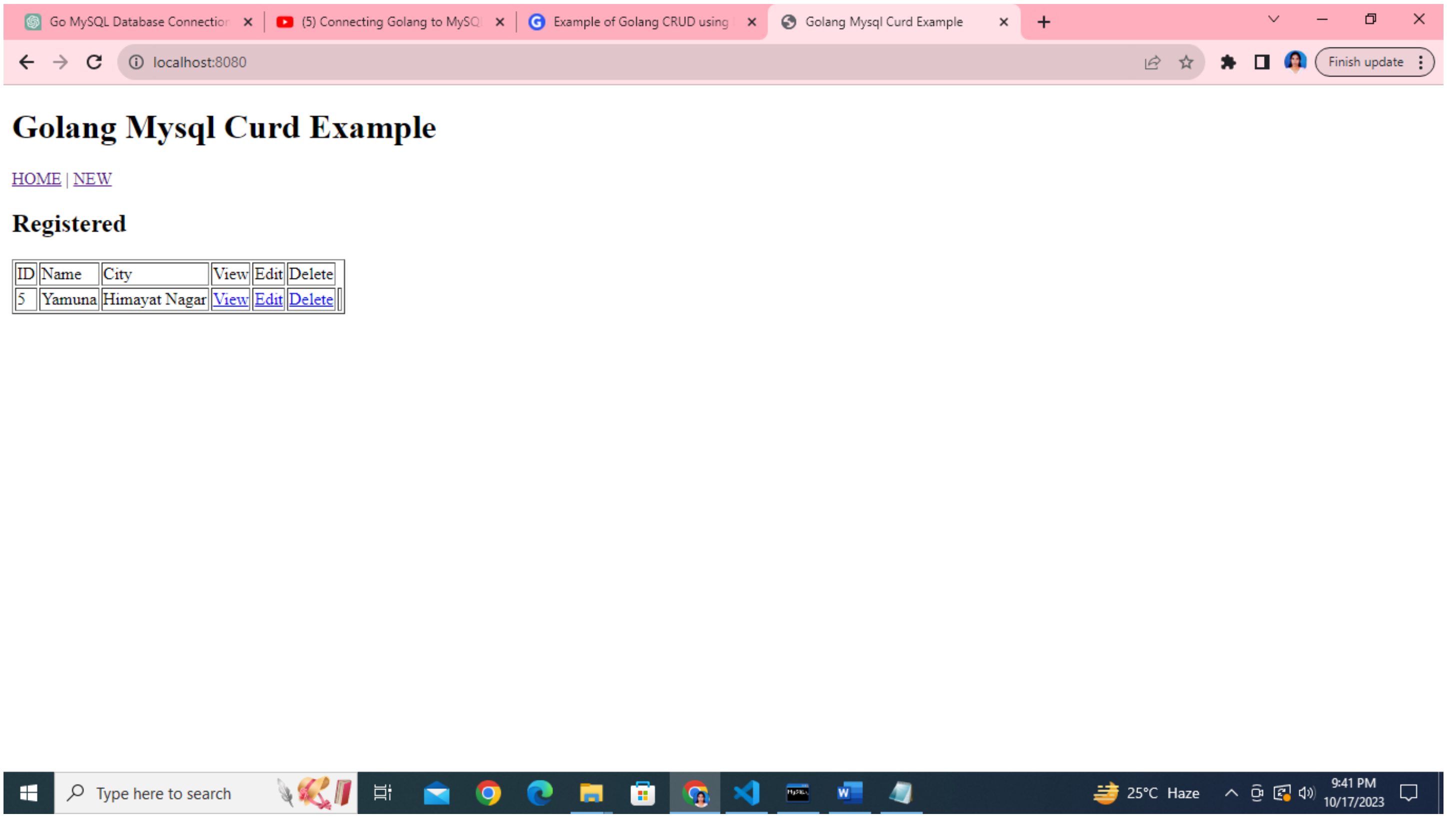
Golang Mysql Curd Example

HOME | NEW

New Name and City

Name Yamuna
City Himayat Nagar
Save user

Fill the data and click on **Save user**


Go MySQL Database Connectior
(5) Connecting Golang to MySQ
Example of Golang CRUD using
Golang Mysql Curd Example
localhost:8080
Finish update
Type here to search
25°C Haze
9:41 PM
10/17/2023

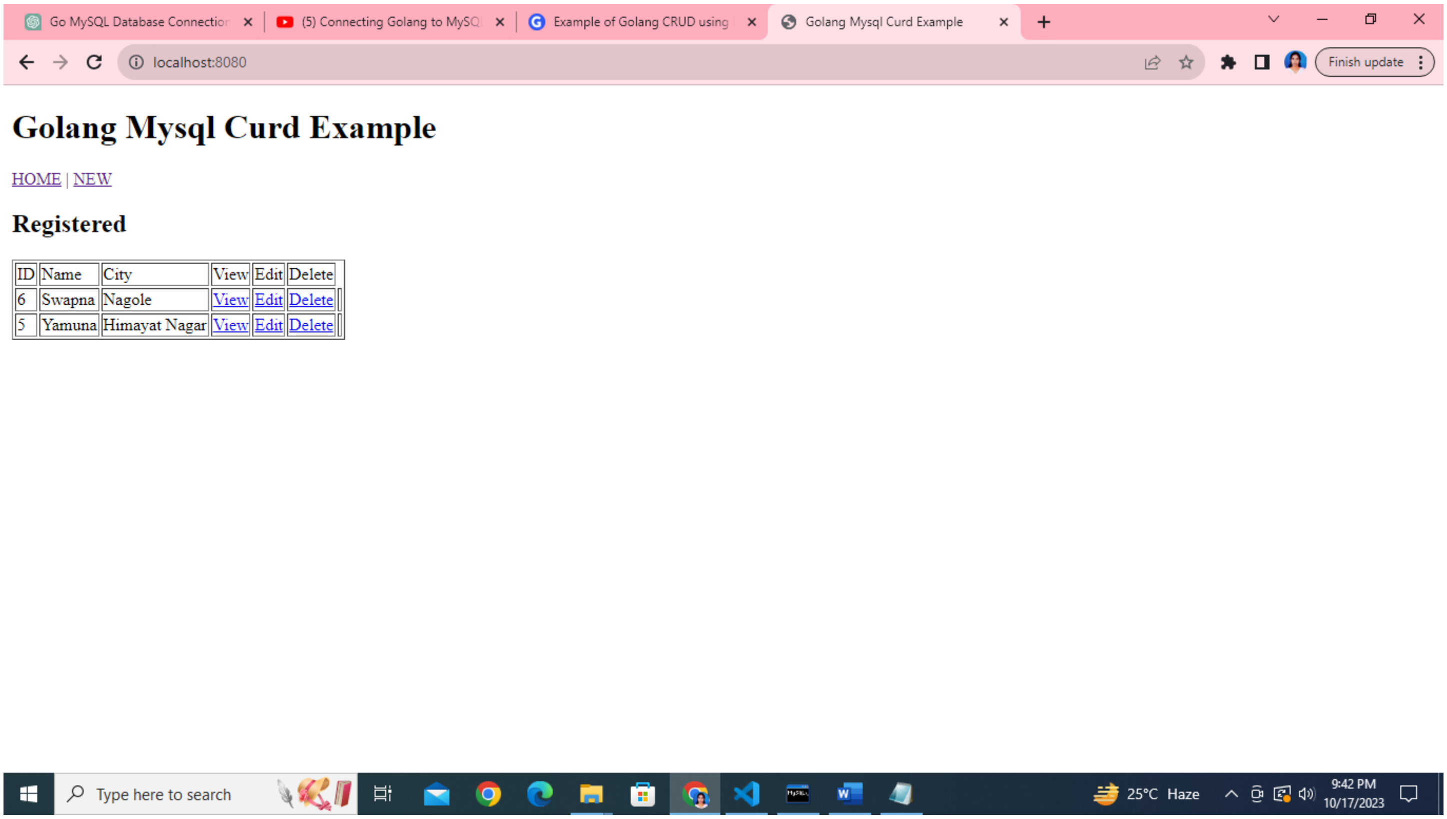
Golang Mysql Curd Example

HOME | NEW

Registered

ID	Name	City	View	Edit	Delete
5	Yamuna	Himayat Nagar	View	Edit	Delete

Again click on New, give some more data


Go MySQL Database Connectio
(5) Connecting Golang to MySQ
Example of Golang CRUD using
Golang Mysql Curd Example
localhost:8080
Finish update
Type here to search
25°C Haze
9:42 PM
10/17/2023

Golang Mysql Curd Example

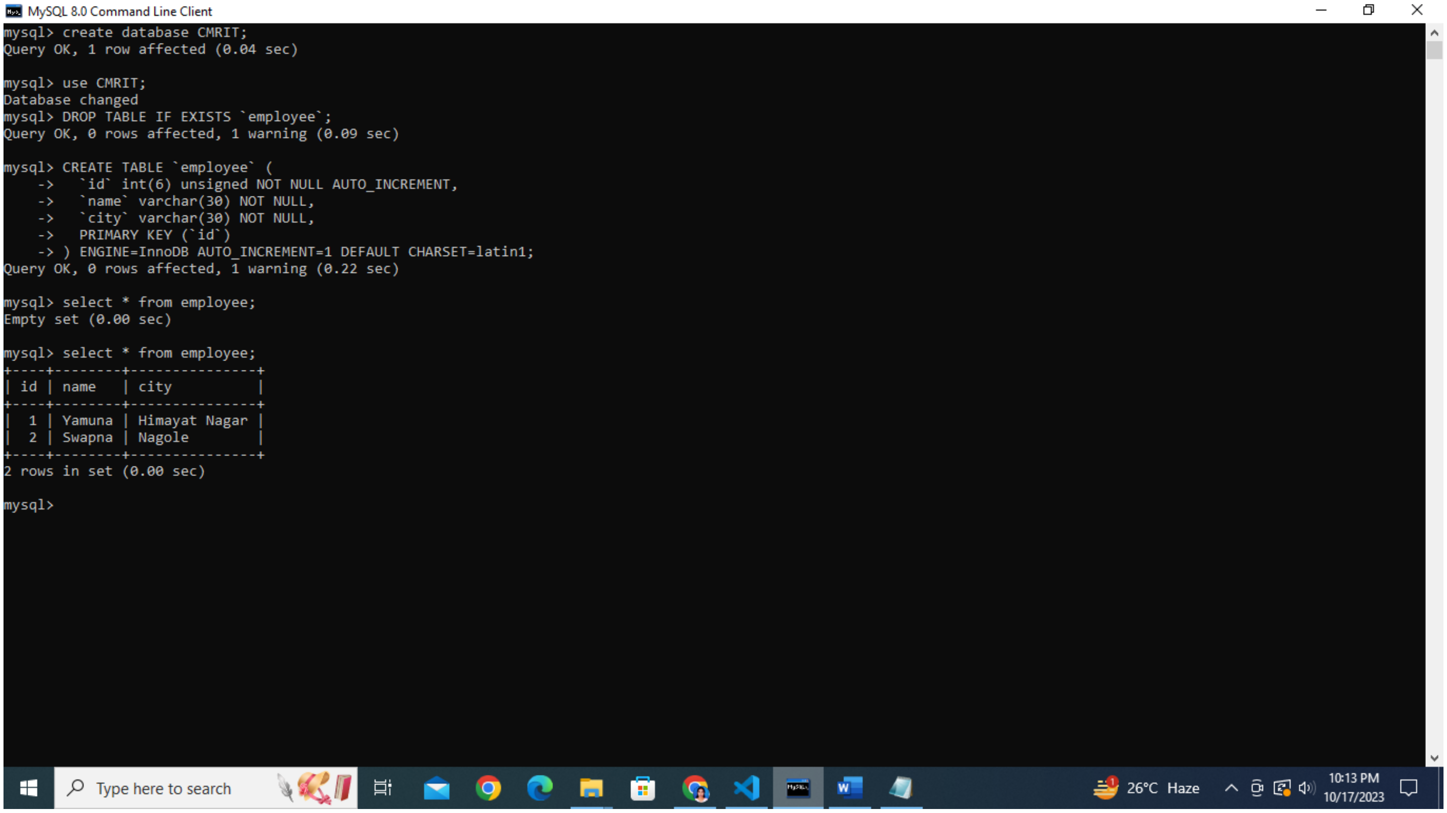
HOME | NEW

Registered

ID	Name	City	View	Edit	Delete
6	Swapna	Nagole	View	Edit	Delete
5	Yamuna	Himayat Nagar	View	Edit	Delete

Now we can **View** the data, **Edit** the data and **Delete** the data.

At last we can check whether our updated data is appeared in the database or not by using **MySQL**.



```
mysql> create database CMRIT;
Query OK, 1 row affected (0.04 sec)

mysql> use CMRIT;
Database changed
mysql> DROP TABLE IF EXISTS `employee`;
Query OK, 0 rows affected, 1 warning (0.09 sec)

mysql> CREATE TABLE `employee` (
    ->   `id` int(6) unsigned NOT NULL AUTO_INCREMENT,
    ->   `name` varchar(30) NOT NULL,
    ->   `city` varchar(30) NOT NULL,
    ->   PRIMARY KEY (`id`)
    -> ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;
Query OK, 0 rows affected, 1 warning (0.22 sec)

mysql> select * from employee;
Empty set (0.00 sec)

mysql> select * from employee;
+----+--------+---------------+
| id | name   | city          |
+----+--------+---------------+
|  1 | Yamuna | Himayat Nagar |
|  2 | Swapna | Nagole        |
+----+--------+---------------+
2 rows in set (0.00 sec)

mysql>
```

Go Googlecrome, type mysql,   Mysql.com

Downloads come down click “Mysql community (GPL) downloads”

select My Installer for windows---select community installer (windows x86, 32 -bit)—2nd one

select “No thanks just start my download

go to download open that file.

Choose the set up type as full, and wait for installation----give strong password.

Type mySQL password: **Yamuna@123** and repeat the same

Gorough the installation, click finish, browser will opened

Mysql JS> and one more browser is welcome to **mysqlbench**

Come to MySql Connections, click on it

Its asking password, give Yamuna@123, save password vault as ok,

Now close all windows

Come to C:/Program Files/MySql/MySQl Server 8.0/bin

----copy the path, come to environment variables (Edit the system environment variables)

Select **Environment variable** come to system variables select **Path** click on new paste the path click on ok ok ok

Open cmd>

```
c:/users/dell/Yamunadevi>mysql –version
```

Then c:/users/dell/Yamunadevi>mysql -u root -p

Enter password: Yamuna@123

Mysql> Mysql> **use CMRIT;** //Now we are in CMRIT database

Create table Employee by using below command

```
Mysql> DROP TABLE IF EXISTS `employee`;
CREATE TABLE `employee` (
  `id` int(6) unsigned NOT NULL AUTO_INCREMENT,
  `name` varchar(30) NOT NULL,
  `city` varchar(30) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;
```

To compile

```
C:\Users\DELL\Desktop\YamunaDB>go get -u github.com/go-sql-driver/mysql .
```

Required files are loaded.

```
C:\Users\DELL\Desktop\YamunaDB>go mod init YamunaDB
C:\Users\DELL\Desktop\YamunaDB>go run main.go
```

Click allow then go to browser and type, http://localhost:8080