**Name : Nithish Kumar A**

## MedApp (GoApi)

**4. Database Table Specification :-**

**4.1Table Definitions:-**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **4.1.1 User** |  |  |  |  |  |
| **ColumnName** | **Data Type** | **Constraint** | **Not Null** | **Auto Increment** | **Possible Values** |
| User\_Id | Int | Primary | yes | yes |  |
| User | Varchar(20) |  |  |  |  |
| Password | Varchar(20) |  |  |  |  |
| Role | Varchar(20) |  |  |  |  |
| Created By | Varchar(50) |  |  |  |  |
| Created Date | DateTime |  |  |  |  |
| Updated By | Varchar(50) |  |  |  |  |
| Updated Date | DateTime |  |  |  |  |
|  |  |  |  |  |  |
| **4.1.2 Stock** |  |  |  |  |  |
| **ColumnName** | **Data Type** | **Constraint** | **Not Null** | **Auto Increment** | **Possible Values** |
| Stock\_Id | Int | Primary | yes | yes |  |
| Med\_Id | Int |  |  |  |  |
| MedicineBrand | Varchar(20) |  |  |  |  |
| Qty | Int |  |  |  |  |
| Price | Int |  |  |  |  |
| Created By | Varchar(50) |  |  |  |  |
| Created Date | DateTime |  |  |  |  |
| Updated By | Varchar(50) |  |  |  |  |
| Updated Date | DateTime |  |  |  |  |
|  |  |  |  |  |  |
| **4.1.3 Medicinemaster** |  |  |  |  |  |
| **ColumnName** | **Data Type** | **Constraint** | **Not Null** | **Auto Increment** | **Possible Values** |
| Med\_Id | Int | Primary | yes | yes |  |
| MedicineName | Varchar(20) |  |  |  |  |
| MedicineBrand | Varchar(20) |  |  |  |  |
| Created By | Varchar(50) |  |  |  |  |
| Created Date | DateTime |  |  |  |  |
| Updated By | Varchar(50) |  |  |  |  |
| Updated Date | DateTime |  |  |  |  |
|  |  |  |  |  |  |
| **4.1.4 Login History** |  |  |  |  |  |
| **ColumnName** | **Data Type** | **Constraint** | **Not Null** | **Auto Increment** | **Possible Values** |
| Login\_id | Int | Primary | yes | yes |  |
| UserId | Int |  |  |  |  |
| Login\_Time | DateTime |  |  |  |  |
| Logout\_Time | DateTime |  |  |  |  |
| Created By | Varchar(50) |  |  |  |  |
| Created Date | DateTime |  |  |  |  |
| Updated By | Varchar(50) |  |  |  |  |
| Updated Date | DateTime |  |  |  |  |
|  |  |  |  |  |  |
| **4.1.5 Bill Master** |  |  |  |  |  |
| **ColumnName** | **Data Type** | **Constraint** | **Not Null** | **Auto Increment** | **Possible Values** |
| BM\_Id | Int | Primary | yes | yes |  |
| BillDate | Date |  |  |  |  |
| BillAmt | Int |  |  |  |  |
| BillGst | Int |  |  |  |  |
| NetPrice | Int |  |  |  |  |
| User\_Id | Int |  |  |  |  |
| Created By | Varchar(50) |  |  |  |  |
| Created Date | DateTime |  |  |  |  |
| Updated By | Varchar(50) |  |  |  |  |
| Updated Date | DateTime |  |  |  |  |
|  |  |  |  |  |  |
| **4.1.6 Bill Detail** |  |  |  |  |  |
| **ColumnName** | **Data Type** | **Constraint** | **Not Null** | **Auto Increment** | **Possible Values** |
| Bill\_Id | Int | Primary | yes | yes |  |
| Med\_Id | Int |  |  |  |  |
| Qty | Int |  |  |  |  |
| UnitPrice | Int |  |  |  |  |
| Amt | Int |  |  |  |  |
| Created By | Varchar(50) |  |  |  |  |
| Created Date | DateTime |  |  |  |  |
| Updated By | Varchar(50) |  |  |  |  |
| Updated Date | DateTime |  |  |  |  |

**5. Database Query Specification :-**

**5.1. Database - SQL Queries :-**

**5.1.1 Query 1**

|  |
| --- |
| **Query parameters: Applicable(Username,Password)** |
| Select User\_Id,UserName,Role From User Where User = ? and Password = ? ; |

**5.1.2 Query 2**

|  |
| --- |
| **Query parameters: Applicable(UserId)** |
| // used Db Function **log\_entry**  select log\_entry(?) |

**5.1.3 Query 3**

|  |
| --- |
| **Query parameters: Not applicable** |
| // For Total inventory value **manager Dashboard**  Select SUM(s.Qty \* s.Price) as Total From stock s; |

**5.1.4 Query 4**

|  |
| --- |
| **Query parameters: Not applicable** |
| // For **Biller** Daily sales **Dashboard**  select (case when c.date\_wise = curdate() then 'Today' else 'yesterday' end)days,nvl(a.amount,'0') amount  from(  select bm.BillDate bill,sum(bm.BillAmt) amount,u.User\_Id  from billmaster bm , userdata u  where bm.User\_id = u.User\_Id and u.`User` = 'Nethaji'  group by bm.BillDate )a  right join  (  SELECT date\_sub(curdate(),interval n day) AS date\_wise  FROM (  SELECT 0 n UNION SELECT 1  )numbers  ) c  on a.bill = c.date\_wise |
|
|

**5.1.5 Query 5**

|  |
| --- |
| **Query parameters: Not applicable** |
| // For Todays sales in **manager Dashboard**  Select SUM(b.BillAmt) From billmaster b,userdata u Where u.User\_Id = b.User\_Id and b.BillDate = date(now()); |
|
|

|  |
| --- |
| **Query parameters: Not applicable** |
| // For WeekWise Line Chart **manager Dashboard**  select dayname(date\_wise), nvl(sum(b.BillAmt), '0') bill\_amount from weekwise w left join billmaster b   on w.date\_wise = b.BillDate  group by date\_wise |
|
|

**5.1.6 Query 6**

**5.1.7 Query 7**

|  |
| --- |
| **Query parameters: Not applicable** |
| // For Monthly Sales Trend manager dashboard  select mw.mnth finalMnth,nvl(b.totalSale, '0') FinalAmt from monthwise mw left join ( select a.months resmnth, sum(a.amount) totalSale from (select monthname(b.BillDate) months,b.BillAmt amount from billmaster b  where b.BillDate between date\_sub(curdate(),interval 1 year) and curdate() ) a  group by a.months )b on mw.mnth = b.resmnth; |
|
|

**5.1.8 Query 8**

|  |
| --- |
| **Query parameters: Not applicable** |
| // For today salesman performance **Pie chart** :  select a.`user`,nvl(b.Amt,'0') amount  from ( select u.User\_Id as id,u.`User`  from userdata u  where u.`Role` ='biller')a left join ( select bm.User\_Id as UserId ,sum(bm.BillAmt) Amt from billmaster bm where bm.BillDate = curdate() group by bm.user\_id)b on a.id=b.UserId |
|
|

**5.1.9 Query 9**

|  |
| --- |
| **Query parameters: Not applicable** |
| // For current month salesman performance **Bar chart**:  select a.`User`,nvl(b.Amt,'0') amount  from ( select u.User\_Id as id, u.`User`  from userdata u where u.`Role` = 'biller')a left join ( select bm.User\_id as UserId ,sum(bm.BillAmt) Amt from billmaster bm where bm.bill\_date between DATE\_FORMAT(NOW(), '%Y-%m-01') and date\_add( DATE\_FORMAT(NOW(), '%Y-%m-01'),interval 1 month) group by mbm.user\_id)b on a.id = b.UserId |
|
|

|  |
| --- |
| **Query parameters: Applicable(MedName,Brand,Username/UserId)** |
| // Use Func query for **med\_entry**  select med\_entry(?,?,?) |

**5.1.10 Query 10**

**5.1.11 Query 11**

|  |
| --- |
| **Query parameters: Applicable(UserName,Password,Role,createdby)** |
| // use Func query for **add\_user**  select add\_user(?,?,?,?) |

**5.1.12 Query 12**

|  |
| --- |
| **Query parameters: Not applicable** |
| // To get **Stock view** dataTable  select s.Med\_id ,s.MedicineBrand ,s.Qty ,s.Price  from stock s |
|
|

**5.1.13 Query 13**

|  |
| --- |
| **Query parameters: Not applicable** |
| // To get **Login History** dataTable  select u.`User` ,date(lh.Login\_Time) login\_date,time(lh.Logout\_Time) login\_time, nvl(date(lh.Logout\_Time),'Working..')logout\_date,nvl(time(lh.Logout\_Time),'Working..')  logout\_time  from loginhistory lh ,userdata u  where u.User\_Id = lh.UserId |
|
|

**5.1.14 Query 14**

|  |
| --- |
| **Query parameters: Applicable(From date,To date)** |
| // For **Sales report** dataTable  select bd.Bill\_No ,date(bm.BillDate) Dates ,mm.MedicineName ,bd.Qty ,bd.Amt  from billdetails bd ,billmaster bm ,medicinemaster mm where bd.Bill\_Id = bm.Bill\_No and mm.Med\_Id = bd.Med\_id and bm.BillDate between date(?) and date(?); |

**5.1.15 Query 15**

|  |
| --- |
| **Query parameters: Not applicable** |
| // To **get Medicine master**   select m.Med\_Id,m.MedicineName,m.MedicineBrand from lmedicinemaster m |
|
|

**5.2. Database - SQL Functions :-**

**5.2.1 Query Func 1**

|  |
| --- |
| **Function for insert and update in stock entry** |
| create function stockEntry(user\_id int,med\_id int,qty int,price int) returns int begin IF EXISTS (SELECT \* FROM stock s WHERE s.Med\_Id = med\_id)then UPDATE stock s SET s.Qty = s.Qty + qty,s.Price = price, s.UpdatedBy = user\_id,s.UpdatedDate = now() WHERE s.Med\_Id = med\_id; return 1; ELSE INSERTINTOstock (Med\_Id,Qty,Price,CreatedBy,CreatedDate) VALUES (med\_id,qty,price,user\_id,now()); return 0; end if; end; |

**5.2.2 Query Func 2**

|  |
| --- |
| **Function for add\_user entry function** |
| create function add\_user (user\_name varchar(255) , password varchar(255), user\_role varchar(255),created\_by varchar(255)) RETURNS varchar(255) begin IF EXISTS (SELECT \* FROM userdata u WHERE u.`User` = user\_name and u.Password = password and u.`Role` = user\_role)then RETURN 'error'; ELSE INSERT INTO userdata (`User`,Password,`Role` ,CreatedBy ,CreatedDate ,UpdatedBy ,UpdatedDate) VALUES (user\_name,password,user\_role,created\_by,now(),created\_by,now()); RETURN 'success'; end if; end; |

**5.2.3 Query Func 3**

|  |
| --- |
| **Function for new medicine master entry** |
| create function med\_entry( med\_name varchar(255),brand varchar(255),username varchar(255)) RETURNS varchar(255) begin  if exists (select \* from medicinemaster mm where mm.MedicineName = med\_name and   mm.Brand = brand) then  RETURN 'MedicineName & Brand Already Exist'; else  insert into medicinemaster (MedicineName,Brand, CreatedBy,CreatedDate,UpdatedBy,UpdatedDate)  values(med\_name,brand,username,now(),username,now()); RETURN 'MedicineName & Brand Added Successfully'; end if; end; |

**5.2.4 Query Func 4**

|  |
| --- |
| **Function for login/logout entry** |
| create function log\_entry( user\_id int) RETURNS varchar(255) begin if exists (select \* from loginhistory lh where lh.UserId = user\_id and lh.Logout\_Time is null) then update loginhistory lh set lh.UpdatedBy = user\_id , lh.UpdatedDate = now() ,  lh.Logout\_Time = now() where lh.UserId = user\_id and lh.Logout\_Time is null ; RETURN 'Updated\_successfully'; else insert into loginhistory (UserId ,Login\_Time ,CreatedBy ,CreatedDate) values (user\_id,now(),user\_id,now()); RETURN 'Insert\_Successfully'; end if; end; |

**5.3 DataBase SQL Views:-**

**5.3.1 View 1**

|  |
| --- |
| **View for Month wise** |
| create view monthwise as (select monthname( DATE\_FORMAT(DATE\_SUB(DATE\_FORMAT(NOW(), '%Y-%m-01'), INTERVAL n MONTH), '%Y-%m-01')) as mnth FROM (SELECT 11 n UNION SELECT 10 UNION SELECT 9 UNION SELECT 8   UNION SELECT 7 UNION SELECT 6 UNION SELECT 5 UNION SELECT 4   UNION SELECT 3 UNION SELECT 2 UNION SELECT 1 UNION SELECT 0) numbers); |

**5.3.2 View 2**

|  |
| --- |
| **View for weekwise** |
| create view weekwise as (SELECT date\_sub(curdate(),interval n day) AS date\_wise FROM (SELECT 0 n UNION SELECT 1 UNION SELECT 2 UNION SELECT 3   UNION SELECT 4 UNION SELECT 5 UNION SELECT 6) numbers); |

**6. GoLang Program Specification :-**

**6.1 Structures Definitions :**

**Login/Userdata Api**

**6.1.1 Login\_data**

|  |  |  |
| --- | --- | --- |
| **Structure Name** | **Login\_data** | |
| Attributes | Data Type | json values |
| UserName | string | `username` |
| Password | string | `password` |

**6.1.2 Login\_resp**

|  |  |  |
| --- | --- | --- |
| **Structure Name** | **Login\_resp** | |
| Attributes | Data Type | json values |
| UserId | int | `user\_id` |
| Name | string | `username` |
| Role | string | `role` |

**6.1.4 Login\_result**

|  |  |  |
| --- | --- | --- |
| **Structure Name** | **Login\_result** | |
| Attributes | Data Type | json values |
| LogInData | Login\_resp | `loginData` |
| Status | string | `status` |
| ErrMsg | string | `errMsg` |

**Get MedMaster Api**

**6.1.5 Master**

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure Name** | **Master** | | |
| Attributes | Data Type | | json values |
| Id | int | `med\_id` | |
| MedName | string | `medName` | |
| Brand | string | `brand` | |

**6.1.6 Master\_resp**

|  |  |  |
| --- | --- | --- |
| **Structure Name** | **Master\_resp** | |
| Attributes | Data Type |  |
| MasterArr | []Master | `masterArr` |
| Status | string | `status` |
| ErrMsg | string | `errMsg` |

**Get Stock Api**

**6.1.7 Stock**

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure Name** | **Stock** | | |
| Attributes | Data Type | json values | |
| MedId | string | | `med\_id` |
| MedicineName | string | | `medName` |
| MedicineBrand | string | | `brand` |
| Qty | float | | `qty` |
| Price | float | | `price` |

**6.1.8 Stock\_resp**

|  |  |  |
| --- | --- | --- |
| **Structure Name** | **Stock\_resp** | |
| Attributes | Data Type | json values |
| Stockrr | []Stock | `stockArr` |
| Status | string | `status` |
| ErrMsg | string | `errMsg` |

**Get Login History Api**

**6.1.9 Loghis**

|  |  |  |
| --- | --- | --- |
| **Structure Name** | **Loghis** | |
| Attributes | Data Type | json values |
| UserName | string | `username` |
| LoginDate | string | `login\_date` |
| LoginTime | string | `login\_time` |
| LogoutDate | string | `logout\_date` |
| LogoutTime | string | `logout\_time` |

**6.1.10 Loghis\_resp**

|  |  |  |
| --- | --- | --- |
| **Structure Name** | **Loghis\_resp** | |
| Attributes | Data Type | json values |
| LoghisArr | []Loghis | `loghisArr` |
| Status | string | `status` |
| ErrMsg | string | `errMsg` |

**Add User Api**

**6.1.11 AddUser**

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure Name** | **AddUser** | | |
| Attributes | Data Type | | json values |
| UserId | int | `user\_id` | |
| UserName | string | `username` | |
| Password | string | `password` | |
| Role | string | `role` | |

**6.1.12 Check\_user**

|  |  |  |
| --- | --- | --- |
| **Structure Name** | **Check\_user** | |
| Attributes | Data Type | json values |
| UserName | string | `username` |
| Name | string | `username` |
| Role | string | `role` |

**6.1.13 AddUser\_resp**

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure Name** | **AddUser\_resp** | | |
| Attributes | Data Type | | json values |
| Status | string | `status` | |
| ErrMsg | string | `errMsg` | |

**Biller Dashboard Api**

**6.1.14 Billdash**

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure Name** | **Billdash** | | |
| Attributes | Data Type | | json values |
| UserId | int | `user\_id` | |

**6.1.15 Billdash\_resp**

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure Name** | **Billdash\_resp** | | |
| Attributes | Data Type | | json values |
| Day | string | `day` | |
| Amount | float | `amt` | |

**6.1.16 Billdash\_result**

|  |  |  |
| --- | --- | --- |
| **Structure Name** | **Billdash\_result** | |
| Attributes | Data Type | json values |
| Billdash\_res | []Billdash | `billerRes` |
| Status | string | `status` |
| ErrMsg | string | `errMsg` |

**Manager Dashboard Api**

**6.1.17 TodaySale**

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure Name** | **TodaySale** | | |
| Attributes | Data Type | | json values |
| Todaysale | float | `todaysale` | |

**6.1.18 TotalInventry**

|  |  |  |
| --- | --- | --- |
| **Structure Name** | **TotalInventry** | |
| Attributes | Data Type | json values |
| Totalinventry | float | `totalInventry` |

**6.1.19 DailySale**

|  |  |  |
| --- | --- | --- |
| **Structure Name** | **DailySale** | |
| Attributes | Data Type | json values |
| Dailysale | string | `dailySale` |

**6.1.20 MonthlySale**

|  |  |  |
| --- | --- | --- |
| **Structure Name** | **MonthlySale** | |
| Attributes | Data Type | json values |
| MonthlySale | string | `monthlySale` |

**6.1.21 TodaySalePerf**

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure Name** | **TodaySalePerf** | | |
| Attributes | Data Type | json values | |
| TodaySalePerf | string | | `todaySalePerf` |

**6.1.22 CurrentMnthPerf**

|  |  |  |
| --- | --- | --- |
| **Structure Name** | **CurrentMnthPerf** | |
| Attributes | Data Type | json values |
| CurrentMnthPerf | string | `currentMnthPerf` |

**6.1.23 Combine**

|  |  |  |
| --- | --- | --- |
| **Structure Name** | **Combine** | |
| Attributes | Data Type | json values |
| TodaysaleVal | Todaysale | `todaysaleVal` |
| TotalinventryVal | Totalinventry | `totalInventryVal` |
| DailysaleArr | []Dailysale | `dailysaleArr` |
| MonthlySaleArr | []MonthlySale | `monthlySaleArr` |
| TodaySalePerfArr | []TodaySalePerf | `todaySalePerfArr` |
| CurrentMnthPerArr | []CurrentMnthPer | `CurrentMnthPerArr` |

**6.1.24 Manager\_resp**

|  |  |  |
| --- | --- | --- |
| **Structure Name** | **Manager\_resp** | |
| Attributes | Data Type | json values |
| Manager\_respArr | []Combine | `manager\_resp` |
| Status | string | `status` |
| ErrMsg | string | `errMsg` |

**Add Medicine Master Api**

**6.1.25 AddMedicine**

|  |  |  |
| --- | --- | --- |
| **Structure Name** | **AddMedicine** | |
| Attributes | Data Type | json values |
| UserId | int | `user\_id` |
| MedicineName | string | `medName` |
| Brand | string | `brand` |

**6.1.26 AddMedicine\_resp**

|  |  |  |
| --- | --- | --- |
| **Structure Name** | **AddMedicine\_resp** | |
| Attributes | Data Type | json values |
| Status | string | `status` |
| ErrMsg | string | `errMsg` |

**Stock Entry Api**

**6.1.27 StockEntry**

|  |  |  |
| --- | --- | --- |
| **Structure Name** | **StockEntry** | |
| Attributes | Data Type | json values |
| UserId | int | `user\_id` |
| MedId | int | `med\_id` |
| Brand | string | `brand` |
| Qty | int | `qty` |
| Price | float | `price` |

**6.1.28 StockEntry\_resp**

|  |  |  |
| --- | --- | --- |
| **Structure Name** | **StockEntry\_resp** | |
| Attributes | Data Type | json values |
| Status | string | `status` |
| ErrMsg | string | `errMsg` |

**Bill Entry Api**

**6.1.30 BillEntry**

|  |  |  |
| --- | --- | --- |
| **Structure Name** | **BillEntry** | |
| Attributes | Data Type | json values |
| UserId | int | `user\_id` |
| MedId | int | `med\_id` |
| BillNo | int | `billNo` |
| Qty | float | `qty` |
| Unitprice | float | `unitPrice` |
| BillAmount | float | `billAmt` |

**6.1.33 BillEntry\_resp**

|  |  |  |
| --- | --- | --- |
| **Structure Name** | **BillEntry\_resp** | |
| Attributes | Data Type | json values |
| Status | string | `status` |
| ErrMsg | string | `errMsg` |

**Sales Report Api**

**6.1.34 SalesReport**

|  |  |  |
| --- | --- | --- |
| **Structure Name** | **BillEntry\_resp** | |
| Attributes | Data Type | json values |
| FromDate | string | `from` |
| ToDate | string | `to` |

**6.1.35 SalesReport\_result**

|  |  |  |
| --- | --- | --- |
| **Structure Name** | **BillEntry** | |
| Attributes | Data Type | json values |
| BillNo | int | `billNo` |
| BillDate | string | `billDate` |
| MedicineName | string | `medName` |
| Qty | float | `qty` |
| BillAmount | float | `billAmt` |

**6.1.36 SalesReport\_res**

|  |  |  |
| --- | --- | --- |
| **Structure Name** | **BillEntry\_resp** | |
| Attributes | Data Type | json values |
| SalesReportArr | []SalesReport\_result | `salesReportArr` |
| Status | string | `status` |
| ErrMsg | string | `errMsg` |

**Logout Api**

**6.1.37 Logout**

|  |  |  |
| --- | --- | --- |
| **Structure Name** | **Logout** | |
| Attributes | Data Type | json values |
| UserId | int | `user\_id` |

**6.1.38 Logout\_res**

|  |  |  |
| --- | --- | --- |
| **Structure Name** | **Logout\_resp** | |
| Attributes | Data Type | json values |
| Status | string | `status` |
| ErrMsg | string | `errMsg` |

**6.2 Methods Definitions :**

**LocalDBConnection**

**6.2.1 LocalDBConnect ()**

|  |  |
| --- | --- |
| **Method Name** | **LocalDBConnect** |
| **Description : This method is used to establish the connection btwn the Api to Database** | |
| **Parameters** | **Not Applicable** |
| **Return Values** | **db,err** |
| **Psuedocode** | |
| 1. LocalDBConnect method is used to establish the connection between the Database and the Golang 2. **Note :-** Use defer db.Close() method to close the connect when the need is over. | |

**User Login + Login entry**

**6.2.2 GetLogin ()**

|  |  |
| --- | --- |
| **Method Name** | **GetLogin** |
| **Description : This method is used to establish the connection btwn the Api to Database** | |
| **Request** | **Login\_data(**body**)** |
| **Return Values** | **Login\_result** |
| **Psuedocode** | |
| 1. Declare a variable type of Login\_data as **data** 2. Declare a variable type of Login\_result as **resp** 3. Ready Body and Unmarshal it. 4. create a structure for the unmarshal values. 5. Call **LocalDBConnect ()** to open the Database 6. Use the SQL **Query 1** 7. Check the given Struct value = To Query return value. 8. if the values are same Make a response as “**E**” 9. if not assign the values to the **Login\_resp** structure and give the user\_id as input to **Query 2**. 10. after that Marshal **resp** and pass the response to Vue js. | |

**Logout entry**

**6.2.3 Logout ()**

|  |  |
| --- | --- |
| **Method Name** | **Logout** |
| **Description : This method is used to establish the connection btwn the Api to Database** | |
| **Parameters** | **Log\_out** |
| **Return Values** | **Logout\_resp** |
| **Psuedocode** | |
| 1. If user presses the **sure btn** in vue the method carries the user\_id 2. Call **LocalDBConnect ()** to open the Database. 3. Declare a variable type of Log\_out as logout. 4. Declare a variable type of Logout\_resp as resp. 5. Pass the user\_id as input to the SQL **Query 2 .** 6. After that Marshal **resp** and pass the status to Vue js. | |

**Login History**

**6.2.4 GetLoginhistory ()**

|  |  |
| --- | --- |
| **Method Name** | **GetLoginhistory** |
| **Description : This method is used to establish the connection btwn the Api to Database** | |
| **Parameters** | **Not Applicable** |
| **Return Values** | **Loghis\_resp** |
| **Psuedocode** | |
| 1. This method is call when the Admin opens the Loginhistory router 2. Declare a variable of type Loghis as loghis. 3. Declare a variable of type Loghis\_resp as resp. 4. Call **LocalDBConnect ()** to open the Database. 5. Use **Query 13** and store the given values to Loghis struct. 6. After that Marshal **resp** and pass the response to Vue js. | |

**Dropdown MedicineMaster**

**6.2.5 GetMedMas ()**

|  |  |
| --- | --- |
| **Method Name** | **GetMedMas** |
| **Description : This method is used to get the records from the medicinemaster table.** | |
| **Parameters** | **Not Applicable** |
| **Return Values** | **Master\_resp** |
| **Psuedocode** | |
| 1. This method is called when mounted and it is used for the biller/inventroy use to pick the medicine from the auto complete field. 2. Call **LocalDBConnect ()** to open the Database. 3. Declare the variable of the type Master as medMas. 4. Declare the variable of the type Master\_resp as resp. 5. Use **Query 15**  and store the values in the Master struct. 6. After that Marshal **resp** and pass the response to Vue js. | |

**Stockview**

**6.2.6 GetStock ()**

|  |  |
| --- | --- |
| **Method Name** | **GetStock** |
| **Description : This method is used to establish the connection btwn the Api to Database** | |
| **Parameters** | **Not Applicable** |
| **Return Values** | **Stock\_resp** |
| **Psuedocode** | |
| 1. This method is called when mounted and it is used for the Biller ,Inventroy & Manager to view current stock in the stock table. 2. Call **LocalDBConnect ()** to open the Database. 3. Declare the variable of the type Stock as stock. 4. Declare the variable of the type Stock\_resp as resp. 5. Use **Query 12** to get the data from the stock table in Database and store the values to the Stock struct. 6. After that Marshal **resp** and pass the response to Vue js. | |

**Adduser**

**6.2.7 Adduser ()**

|  |  |
| --- | --- |
| **Method Name** | **Adduser** |
| **Description : This method is used to add new user,password and role in userdata table.** | |
| **Parameters** | **AddUser** |
| **Return Values** | **Adduser\_resp** |
| **Psuedocode** | |
| 1. This method is called when the Admin presses the **Add btn** after fill the details in the Add User dropdown box. 2. Declare the variable of the type Adduser as add. 3. Declare the variable of the type addUser\_resp as resp. 4. Unmarshal the body and store the values from the front to the AddUser struct. 5. Call **LocalDBConnect ()** to open the Database. 6. Use **Query 11** and pass the unmarshaled struct as input to the query. 7. After that Marshal **resp** and pass the response to Vue js. | |

**Biller dashboard**

**6.2.8 GetBillerdash ()**

|  |  |
| --- | --- |
| **Method Name** | **GetBillerdash** |
| **Description : This method is used to establish the connection btwn the Api to Database** | |
| **Parameters** | **Not Applicable** |
| **Return Values** | **db,err** |
| **Psuedocode** | |
| 1. This method is called when the Biller landed on the dashboard screen. 2. Unmarshall the Body and store the value in the Billdash struct. 3. Call **LocalDBConnect ()** to open the Database. 4. Declare the variable of the type Billdash as bill. 5. Declare the variable of the type Billdash\_result as resp. 6. Use **Query 4**  and pass the Body as the parameter and Store the recevied values to the Billdash\_resp struct 7. After that Marshal **resp** and pass the response to Vue js. | |

**BillEntry**

**6.2.9 BillEntry ()**

|  |  |
| --- | --- |
| **Method Name** | **BillEntry** |
| **Description : This method is used to establish the connection btwn the Api to Database** | |
| **Parameters** | **Not Applicable** |
| **Return Values** | **db,err** |
| **Psuedocode** | |
| 1. This method is called when the Biller presses the **save btn** in Bill Entry drop down. 2. Unmarshall the Body and store the value in the Billdash struct. 3. Call **LocalDBConnect ()** to open the Database. 4. Declare the variable of the type BillEntry as billentry. 5. Declare the variable of the type BillEntry\_resp as resp. 6. Use **Query 16** and and pass the Body as the parameter and Store the recevied values to the BillEntry struct. 7. After that Marshal **resp** and pass the response to Vue js. | |

**Manager dashboard**

**6.2.10 GetManagerdash ()**

|  |  |
| --- | --- |
| **Method Name** | **GetManagerdash** |
| **Description : This method is used to establish the connection btwn the Api to Database** | |
| **Parameters** | **Not Applicable** |
| **Return Values** | **db,err** |
| **Psuedocode** | |
|  | |

**SalesReport**

**6.2.11 GetSalesReport ()**

|  |  |
| --- | --- |
| **Method Name** | **GetSalesReport** |
| **Description : This method is used to establish the connection btwn the Api to Database** | |
| **Parameters** | **Not Applicable** |
| **Return Values** | **db,err** |
| **Psuedocode** | |
|  | |

**Add New Medicine**

**6.2.12 AddMedMas ()**

|  |  |
| --- | --- |
| **Method Name** | **AddMedMas** |
| **Description : This method is used to establish the connection btwn the Api to Database** | |
| **Parameters** | **Not Applicable** |
| **Return Values** | **db,err** |
| **Psuedocode** | |
|  | |

**Stock Entry**

**6.2.13 StockEntry ()**

|  |  |
| --- | --- |
| **Method Name** | **StockEntry** |
| **Description : This method is used to establish the connection btwn the Api to Database** | |
| **Parameters** | **Not Applicable** |
| **Return Values** | **db,err** |
| **Psuedocode** | |
|  | |