create database dbFive;

use dbFive;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CREATE TABLE `categoryTest` (

`id` int(11) PRIMARY KEY NOT NULL AUTO\_INCREMENT,

`categoryName` varchar(50) NOT NULL

)ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO categoryTest(categoryName)

VALUES ('Buyers');

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CREATE TABLE `vendorGeneralInformationC` (

`id` int(11) PRIMARY KEY NOT NULL AUTO\_INCREMENT,

`name` varchar(50) NOT NULL,

`isActive` BOOLEAN NOT NULL,

`street` varchar(80) NOT NULL,

`zip` varchar(20) NOT NULL,

`city` varchar(30) NOT NULL,

`state` varchar(30) NOT NULL,

`phone` varchar(20),

`phoneTwo` varchar(20),

`fax` varchar(20),

`contact` varchar(50),

`email` varchar(50) NOT NULL,

`categoryId` int(11) DEFAULT NULL,

`accountNumber` varchar(30),

`defaultGLAccount` varchar(30),

`nameToPrintOnCheckOr1099` varchar(50),

`mailingName` varchar(50),

`mailingStreet` varchar(80),

`mailingCityStateZip` varchar(80),

`socialSecurityNumber` varchar(20),

`isNon1099Vendor` BOOLEAN NOT NULL,

`notes` varchar(400),

FOREIGN KEY (`categoryId`) REFERENCES `categoryTest` (`id`) ON DELETE SET NULL ON UPDATE CASCADE

)ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO vendorGeneralInformationC(name,isActive,street,zip,city,state,email,categoryId,isNon1099Vendor) VALUES

('Krishna',1,'JJ nagar','99','BANGORE','Maine','abcd@gmail.com',1,0);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CREATE TABLE `addingVendor\_BillHistoryB` (

`id` int(11) PRIMARY KEY NOT NULL AUTO\_INCREMENT,

`vendorId` int(11) DEFAULT NULL,

`date` varchar(20) NOT NULL,

`dateDue` varchar(20) NOT NULL,

`datePaid` varchar(20),

`amount` DECIMAL(10,2) NOT NULL,

`invoice` varchar(50) NOT NULL,

`status` varchar(50),

`checkNumber` varchar(50),

`isCleared` BOOLEAN DEFAULT 0,

`checkMemo` varchar(200),

FOREIGN KEY (`vendorId`) REFERENCES `vendorGeneralInformationC` (`id`) ON DELETE SET NULL ON UPDATE CASCADE

)ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO addingVendor\_BillHistoryB(vendorId,date,dateDue,amount,invoice,checkMemo) VALUES (1,'26\_12\_19','27\_12\_19',1000,123,'Invoice number is 123');

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CREATE TABLE `addingVehicle\_GeneralFourteen` (

`stockNumber` int(11) PRIMARY KEY NOT NULL AUTO\_INCREMENT,

`mileage` int(10) DEFAULT 0,

`isExempt` BOOLEAN DEFAULT 0,

`locationCode` varchar(50),

`mileageIs` varchar(50),

`newOrUseOrRebuilt` varchar(20),

`isCertified` BOOLEAN DEFAULT 1,

`vin` varchar(20),

`year` varchar(20),

`transmission` varchar(20),

`transpeed` varchar(20),

`driveTrain` varchar(20),

`make` varchar(20),

`model` varchar(20),

`style` varchar(20),

`bodyType` varchar(20),

`exteriorColor` varchar(20),

`exteriorColorTwo` varchar(20),

`colorDescription` varchar(20),

`interiorColor` varchar(20),

`tag` varchar(20),

`decalNumber` varchar(20),

`county` varchar(30),

`titleApplication` varchar(50),

`stateTitleIn` varchar(30),

`isTitleIn` BOOLEAN DEFAULT 0,

`titleNumber` varchar(50),

`inspectionNumber` varchar(50),

`inspectionDate` varchar(20),

`inspectedBy` varchar(50),

`gpsSerialNumber` varchar(50),

`engineCylinders` varchar(20),

`engineSize` varchar(20),

`vehType` varchar(20),

`vehCondition` varchar(20),

`vehWeight` varchar(20),

`expirationDate` varchar(20),

`fuelType` varchar(20),

`mpgCombined` varchar(20),

`mpgCity` varchar(20),

`mpgHighway` varchar(20),

`ignitionKeyCode` varchar(30),

`doorKeyCode` varchar(30),

`valetKeyCode` varchar(30),

`isInspected` BOOLEAN DEFAULT 0,

`isDuplicateKeys` BOOLEAN DEFAULT 1,

`purchaseDate` varchar(30),

`originalCost` DECIMAL(10,2) DEFAULT '0.00',

`buyersFee` DECIMAL(10,2) DEFAULT '0.00',

`lotFee` DECIMAL(10,2) DEFAULT '0.00',

`addedCosts` DECIMAL(10,2) DEFAULT '0.00',

`floorPlanCost` DECIMAL(10,2) DEFAULT '0.00',

`totalCosts` DECIMAL(10,2) DEFAULT '0.00',

`howDidYouPay` varchar(50),

`vehicleSource` varchar(50) DEFAULT 'Company',

`vendorId` int(11) DEFAULT NULL,

`buyer` varchar(50),

`notes` varchar(400),

`isDraft` BOOLEAN DEFAULT 1,

`draftNumber` varchar(50),

`datePaid` varchar(50),

FOREIGN KEY (`vendorId`) REFERENCES `vendorGeneralInformationC` (`id`) ON DELETE SET NULL ON UPDATE CASCADE

)ENGINE=InnoDB AUTO\_INCREMENT=1000 DEFAULT CHARSET=utf8;

CREATE TABLE `addingVehicle\_GeneralTen` (

`stockNumber` int(11) PRIMARY KEY NOT NULL AUTO\_INCREMENT,

`mileage` int(10) DEFAULT 0,

`isExempt` BOOLEAN DEFAULT 0,

`locationCode` varchar(50),

`mileageIs` varchar(50),

`newOrUseOrRebuilt` varchar(20),

`isCertified` BOOLEAN DEFAULT 1,

`vin` varchar(20),

`year` varchar(20),

`transmission` varchar(20),

`transpeed` varchar(20),

`driveTrain` varchar(20),

`make` varchar(20),

`model` varchar(20),

`style` varchar(20),

`bodyType` varchar(20),

`exteriorColor` varchar(20),

`exteriorColorTwo` varchar(20),

`colorDescription` varchar(20),

`interiorColor` varchar(20),

`tag` varchar(20),

`decalNumber` varchar(20),

`county` varchar(30),

`titleApplication` varchar(50),

`stateTitleIn` varchar(30),

`isTitleIn` BOOLEAN DEFAULT 0,

`titleNumber` varchar(50),

`inspectionNumber` varchar(50),

`inspectionDate` varchar(20),

`inspectedBy` varchar(50),

`gpsSerialNumber` varchar(50),

`engineCylinders` varchar(20),

`engineSize` varchar(20),

`vehType` varchar(20),

`vehCondition` varchar(20),

`vehWeight` varchar(20),

`expirationDate` varchar(20),

`fuelType` varchar(20),

`mpgCombined` varchar(20),

`mpgCity` varchar(20),

`mpgHighway` varchar(20),

`ignitionKeyCode` varchar(30),

`doorKeyCode` varchar(30),

`valetKeyCode` varchar(30),

`isInspected` BOOLEAN DEFAULT 0,

`isDuplicateKeys` BOOLEAN DEFAULT 1,

`purchaseDate` varchar(30),

`originalCost` DECIMAL(10,2) DEFAULT '0.00',

`buyersFee` DECIMAL(10,2) DEFAULT '0.00',

`lotFee` DECIMAL(10,2) DEFAULT '0.00',

`addedCosts` DECIMAL(10,2) DEFAULT '0.00',

`floorPlanCost` DECIMAL(10,2) DEFAULT '0.00',

`totalCosts` DECIMAL(10,2) DEFAULT '0.00',

`howDidYouPay` varchar(50) NULL,

`vehicleSource` varchar(50) DEFAULT 'Company',

`vendorId` int(11) DEFAULT NULL,

`buyer` varchar(50),

`notes` varchar(400),

`isDraft` BOOLEAN DEFAULT 1,

`draftNumber` varchar(50),

`datePaid` varchar(50),

FOREIGN KEY (`vendorId`) REFERENCES `vendorGeneralInformationC` (`id`) ON DELETE SET NULL ON UPDATE CASCADE

)ENGINE=InnoDB AUTO\_INCREMENT=1000 DEFAULT CHARSET=utf8;

INSERT INTO addingVehicle\_GeneralTen(mileage,locationCode,newOrUseOrRebuilt,vin,year,exteriorColor,inspectionDate,inspectedBy,vehType,purchaseDate,originalCost,buyersFee,addedCosts,floorPlanCost,totalCosts,vendorId) VALUES (0,'ME','new','1HGBH41JX1091001','2018','white','2\_10\_2018','Mr InspectorTwo','Hatch Back','2\_10\_2018',16000,200,500,100,17500,1);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CREATE TABLE `addingVendor\_GLPostsOne` (

`id` int(11) PRIMARY KEY NOT NULL AUTO\_INCREMENT,

`date` varchar(30),

`amount` int(11) NOT NULL,

`referenceOrCheckNumber` varchar(30),

`descriptionOne` varchar(400),

`vendorId` int(11) DEFAULT NULL,

FOREIGN KEY (`vendorId`) REFERENCES `vendorGeneralInformationC` (`id`) ON DELETE SET NULL ON UPDATE CASCADE

)ENGINE=InnoDB DEFAULT CHARSET=utf8;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CREATE TABLE `addingVendor\_addedCosts` (

`id` int(11) PRIMARY KEY NOT NULL AUTO\_INCREMENT,

`date` varchar(30),

`cost` DECIMAL(10,2) DEFAULT '0.00',

`vehicleId` int(11) DEFAULT NULL,

`description` varchar(400),

`ticket` varchar(50),

`gl` int(11) DEFAULT NULL,

`vendorId` int(11) DEFAULT NULL,

FOREIGN KEY (`vendorId`) REFERENCES `vendorGeneralInformationC` (`id`) ON DELETE SET NULL ON UPDATE CASCADE,

FOREIGN KEY (`vehicleId`) REFERENCES `addingVehicle\_GeneralTen` (`stockNumber`) ON DELETE SET NULL ON UPDATE CASCADE

)ENGINE=InnoDB DEFAULT CHARSET=utf8;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Now we add 2 triggers.

1.

delimiter #

create trigger addingVehicle\_GeneralTen\_TriggerOne after insert on addingVehicle\_GeneralTen

for each row

BEGIN

insert into addingVendor\_addedCosts(date,cost,vehicleId,description,ticket,gl,vendorId) VALUES (new.purchaseDate,new.addedCosts,new.stockNumber,'vehicle Added costs',uuid(),1,new.vendorId);

end#

Syntax:

mysql> delimiter #

mysql> create trigger addingVehicle\_GeneralTen\_TriggerOne after insert on addingVehicle\_GeneralTen

-> for each row

-> BEGIN

-> insert into addingVendor\_addedCosts(date,cost,vehicleId,description,ticket,gl) VALUES

(new.purchaseDate,new.addedCosts,new.stockNumber,'vehicle Added costs',uuid(),1);

-> end#

Query OK, 0 rows affected (0.10 sec)

mysql> delimiter ;

Testing:

select \* from addingVendor\_addedCosts;

mysql> select \* from addingVendor\_addedCosts;

Empty set (0.00 sec)

Now insert a record into addingVehicle\_GeneralTen.

INSERT INTO addingVehicle\_GeneralTen(mileage,locationCode,newOrUseOrRebuilt,vin,year,exteriorColor,inspectionDate,inspectedBy,vehType,purchaseDate,originalCost,buyersFee,addedCosts,floorPlanCost,totalCosts,vendorId) VALUES (0,'ME','new','1HGBH41JX1091002','2018','white','2\_10\_2018','Mr InspectorTwo','Hatch Back','2\_10\_2018',16000,200,599,100,17599,1);

mysql> INSERT INTO addingVehicle\_GeneralTen(mileage,locationCode,newOrUseOrRebuilt,vin,year,exteriorColor,inspectionDate,inspectedBy,vehType,purchaseDate,originalCost,buyersFee,addedCosts,floorPlanCost,totalCosts,vendorId) VALUES (0,'ME','new','1HGBH41JX1091002','2018','white','2\_10\_2018','Mr InspectorTwo','Hatch Back','2\_10\_2018',16000,200,599,100,17599,1);

Query OK, 1 row affected (0.04 sec)

Lets see the addingVendor\_addedCosts table.

select \* from addingVendor\_addedCosts;

mysql> select \* from addingVendor\_addedCosts;

mysql> select \* from addingVendor\_addedCosts;

+----+-----------+--------+-----------+---------------------+--------------------------------------+------+----------+

| id | date | cost | vehicleId | description | ticket | gl | vendorId |

+----+-----------+--------+-----------+---------------------+--------------------------------------+------+----------+

| 1 | 2\_10\_2018 | 599.00 | 1001 | vehicle Added costs | 824cb94b-2b06-11ea-b7a3-acb57df5eca5 | 1 | 1 |

+----+-----------+--------+-----------+---------------------+--------------------------------------+------+----------+

1 row in set (0.00 sec)

2.

delimiter #

create trigger addingVehicle\_GeneralTen\_TriggerFive after insert on addingVendor\_addedCosts

for each row

BEGIN

set @sumAddedCosts = (select SUM(cost) FROM addingVendor\_addedCosts where vendorId =new.vendorId);

INSERT INTO addingVendor\_GLPostsOne(date,amount,referenceOrCheckNumber,descriptionOne,vendorId)

SELECT \* FROM(SELECT new.date,@sumAddedCosts,'VALUE ADDED COSTS','Description',new.vendorId) AS tmp

WHERE NOT EXISTS (

SELECT vendorId FROM addingVendor\_GLPostsOne WHERE vendorId = new.vendorId

)LIMIT 1;

end#

3.

delimiter #

create trigger addingVehicle\_GeneralTen\_TriggerSix after insert on addingVendor\_addedCosts

for each row

BEGIN

set @sumAddedCosts = (select SUM(cost) FROM addingVendor\_addedCosts where vendorId =new.vendorId);

update addingVendor\_GLPostsOne SET date = new.date, amount = @sumAddedCosts, referenceOrCheckNumber = 'VALUE ADDED COSTS', descriptionOne = 'Description',vendorId =new.vendorId where vendorId =new.vendorId;

end#

After successful execution of the trigger,we close the sql and open it again.

Lets see the addingVendor\_GLPostsOne table.

select \* from addingVendor\_GLPostsOne;

mysql> select \* from addingVendor\_GLPostsOne;

mysql> select \* from addingVendor\_GLPostsOne;

Empty set (0.00 sec)

Now insert a record into addingVehicle\_GeneralTen.

INSERT INTO addingVehicle\_GeneralTen(mileage,locationCode,newOrUseOrRebuilt,vin,year,exteriorColor,inspectionDate,inspectedBy,vehType,purchaseDate,originalCost,buyersFee,addedCosts,floorPlanCost,totalCosts,vendorId) VALUES (0,'ME','new','1HGBH41JX1091002','2018','white','2\_10\_2018','Mr InspectorTwo','Hatch Back','2\_10\_2018',16000,200,599,100,17599,1);

mysql> INSERT INTO addingVehicle\_GeneralTen(mileage,locationCode,newOrUseOrRebuilt,vin,year,exteriorColor,inspectionDate,inspectedBy,vehType,purchaseDate,originalCost,buyersFee,addedCosts,floorPlanCost,totalCosts,vendorId) VALUES (0,'ME','new','1HGBH41JX1091002','2018','white','2\_10\_2018','Mr InspectorTwo','Hatch Back','2\_10\_2018',16000,200,599,100,17599,1);

Query OK, 1 row affected (0.13 sec)

Lets see the addingVendor\_GLPostsOne table.

select \* from addingVendor\_GLPostsOne;

mysql> show triggers;

+--------------------------------------+--------+--------------------------+--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+--------+------------------------+-------------------------------------------------------------------------------------------------------------------------------------------+----------------+----------------------+----------------------+--------------------+

| Trigger | Event | Table | Statement | Timing | Created | sql\_mode | Definer | character\_set\_client | collation\_connection | Database Collation |

+--------------------------------------+--------+--------------------------+--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+--------+------------------------+-------------------------------------------------------------------------------------------------------------------------------------------+----------------+----------------------+----------------------+--------------------+

| addingVehicle\_GeneralTen\_TriggerOne | INSERT | addingVehicle\_GeneralTen | BEGIN

insert into addingVendor\_addedCosts(date,cost,vehicleId,description,ticket,gl,vendorId) VALUES (new.purchaseDate,new.addedCosts,new.stockNumber,'vehicle Added costs',uuid(),1,new.vendorId);

end | AFTER | 2019-12-30 22:10:18.69 | ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION | root@localhost | utf8 | utf8\_general\_ci | latin1\_swedish\_ci |

| addingVehicle\_GeneralTen\_TriggerFive | INSERT | addingVendor\_addedCosts | BEGIN

set @sumAddedCosts = (select SUM(cost) FROM addingVendor\_addedCosts where vendorId =new.vendorId);

INSERT INTO addingVendor\_GLPostsOne(date,amount,referenceOrCheckNumber,descriptionOne,vendorId)

SELECT \* FROM(SELECT new.date,@sumAddedCosts,'VALUE ADDED COSTS','Description',new.vendorId) AS tmp

WHERE NOT EXISTS (

SELECT vendorId FROM addingVendor\_GLPostsOne WHERE vendorId = new.vendorId

)LIMIT 1;

end | AFTER | 2019-12-30 22:33:48.08 | ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION | root@localhost | utf8 | utf8\_general\_ci | latin1\_swedish\_ci |

| addingVehicle\_GeneralTen\_TriggerSix | INSERT | addingVendor\_addedCosts | BEGIN

set @sumAddedCosts = (select SUM(cost) FROM addingVendor\_addedCosts where vendorId =new.vendorId);

update addingVendor\_GLPostsOne SET date = new.date, amount = @sumAddedCosts, referenceOrCheckNumber = 'VALUE ADDED COSTS', descriptionOne = 'Description',vendorId =new.vendorId where vendorId =new.vendorId;

end | AFTER | 2019-12-30 22:38:09.94 | ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION | root@localhost | utf8 | utf8\_general\_ci | latin1\_swedish\_ci |

+--------------------------------------+--------+--------------------------+--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+--------+------------------------+-------------------------------------------------------------------------------------------------------------------------------------------+----------------+----------------------+----------------------+--------------------+

3 rows in set (0.00 sec)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

mysql> show triggers;

+--------------------------------------+--------+-------------------------------+--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+--------+------------------------+-------------------------------------------------------------------------------------------------------------------------------------------+----------------+----------------------+----------------------+--------------------+

| Trigger | Event | Table | Statement | Timing | Created | sql\_mode | Definer | character\_set\_client | collation\_connection | Database Collation |

+--------------------------------------+--------+-------------------------------+--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+--------+------------------------+-------------------------------------------------------------------------------------------------------------------------------------------+----------------+----------------------+----------------------+--------------------+

| **addingVehicle\_GeneralTen\_TriggerFou**r | INSERT | addingVehicle\_GeneralFourteen | BEGIN

set @stNumber = (SELECT stockNumber FROM addingVehicle\_GeneralFourteen ORDER BY stockNumber DESC LIMIT 1);

insert into addingVendor\_addedCosts(date,cost,vehicleId,description,ticket,gl,vendorId) VALUES (new.purchaseDate,new.addedCosts,@stNumber,'vehicle Added costs',uuid(),1,new.vendorId);

end | AFTER | 2020-01-11 12:12:22.65 | ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION | root@localhost | utf8 | utf8\_general\_ci | latin1\_swedish\_ci |

| addingVehicle\_GeneralTen\_TriggerOne | INSERT | addingVehicle\_GeneralTen | BEGIN

insert into addingVendor\_addedCosts(date,cost,vehicleId,description,ticket,gl,vendorId) VALUES (new.purchaseDate,new.addedCosts,new.stockNumber,'vehicle Added costs',uuid(),1,new.vendorId);

end | AFTER | 2020-01-11 13:13:11.43 | ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION | root@localhost | utf8 | utf8\_general\_ci | latin1\_swedish\_ci |

| **addingVehicle\_GeneralTen\_TriggerFive** | INSERT | addingVendor\_addedCosts | BEGIN

set @sumAddedCosts = (select SUM(cost) FROM addingVendor\_addedCosts where vendorId =new.vendorId);

INSERT INTO addingVendor\_GLPostsOne(date,amount,referenceOrCheckNumber,descriptionOne,vendorId)

SELECT \* FROM(SELECT new.date,@sumAddedCosts,'VALUE ADDED COSTS','Description',new.vendorId) AS tmp

WHERE NOT EXISTS (

SELECT vendorId FROM addingVendor\_GLPostsOne WHERE vendorId = new.vendorId

)LIMIT 1;

end | AFTER | 2020-01-11 14:08:58.12 | ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION | root@localhost | utf8 | utf8\_general\_ci | latin1\_swedish\_ci |

| **addingVehicle\_GeneralTen\_TriggerSix** | INSERT | addingVendor\_addedCosts | BEGIN

set @sumAddedCosts = (select SUM(cost) FROM addingVendor\_addedCosts where vendorId =new.vendorId);

update addingVendor\_GLPostsOne SET date = new.date, amount = @sumAddedCosts, referenceOrCheckNumber = 'VALUE ADDED COSTS', descriptionOne = 'Description',vendorId =new.vendorId where vendorId =new.vendorId;

end | AFTER | 2020-01-11 14:09:25.38 | ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION | root@localhost | utf8 | utf8\_general\_ci | latin1\_swedish\_ci |

+--------------------------------------+--------+-------------------------------+--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+--------+------------------------+-------------------------------------------------------------------------------------------------------------------------------------------+----------------+----------------------+----------------------+--------------------+

4 rows in set (0.00 sec)

mysql>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Now the node.js api part:

Now ,the backend.

1.New Project

npm init

2.Install dependencies

npm install --save express@4.17.1 mysql@2.17.1 body-parser@1.19.0

3.Here index.js is the entry point.Below is the file code.

var http = require('http');

var express = require('express');

var app = express();

var bodyParser = require('body-parser');

var mysql = require('mysql');

//npm install flake-idgen

//https://blog.abelotech.com/posts/generate-unique-identifier-nodejs-javascript/

var FlakeIdGen = require('flake-idgen'),

intformat = require('biguint-format'),

generator = new FlakeIdGen()

var id1 = generator.next()

var stockId;

app.use(bodyParser.json());

app.use(bodyParser.urlencoded({

extended: true

}));

var description = '';

//const uuidv1 = require('uuid/v1');

//uuidv1(); // -> '6c84fb90-12c4-11e1-840d-7b25c5ee775a'

var ticket = generator.next();

var gl = 1;

// default route

app.get('/', function (req, res) {

return res.send({ error: true, message: 'hello' })

});

// connection configurations

var dbConn = mysql.createConnection({

host: 'localhost',

user: 'root',

password: 'krishna',

database: 'dbFive'

/\* host: 'localhost',

user: 'root',

password: '',

database: 'node\_js\_api'

\*/

});

// connect to database

dbConn.connect();

//rest api to get all categories

app.get('/categories', function (req, res) {

console.log(req);

dbConn.query('select \* from categoryTest', function (error, results, fields) {

if (error) throw error;

res.end(JSON.stringify(results));

});

});

//rest api to get vendorGeneralInformation

app.get('/vendorGeneralInformation', function (req, res) {

console.log(req);

dbConn.query('select \* from vendorGeneralInformationC', function (error, results, fields) {

if (error) throw error;

res.end(JSON.stringify(results));

});

});

app.get('/addingVehicleGeneralTen', function (req, res) {

console.log(req);

dbConn.query('select \* from addingVehicle\_GeneralTen', function (error, results, fields) {

if (error) throw error;

res.end(JSON.stringify(results));

});

});

app.get('/vendorAddedCosts', function (req, res) {

console.log(req);

dbConn.query('select \* from addingVendor\_addedCosts', function (error, results, fields) {

if (error) throw error;

res.end(JSON.stringify(results));

});

});

app.get('/addingVendorGLPostsOne', function (req, res) {

console.log(req);

dbConn.query('select \* from addingVendor\_GLPostsOne', function (error, results, fields) {

if (error) throw error;

res.end(JSON.stringify(results));

});

});

//rest api to get billHistory

app.get('/VendorBillHistory', function (req, res) {

console.log(req);

dbConn.query('select \* from addingVendor\_BillHistoryB', function (error, results, fields) {

if (error) throw error;

res.end(JSON.stringify(results));

});

});

app.post('/createVendorGeneralInformation', function(req, res, next) {

var name = req.body.name;

var isActive = req.body.isActive;

var street = req.body.street;

var zip = req.body.zip;

var city = req.body.city;

var state = req.body.state;

var phone = req.body.phone;

var phoneTwo = req.body.phoneTwo;

var fax = req.body.fax;

//name,isActive,street,zip,city,state,phone,phoneTwo,fax,contact,email,categoryId,accoutNumber,defaultGLAccount,nameToPrintOnCheckOr1099,mailingName,mailingStreet,mailingCityStateZip,socialSecurityNumber,isNon1099Vendor,notes

var contact = req.body.contact;

var email = req.body.email;

var categoryId = req.body.categoryId;

var accountNumber = req.body.accoutNumber;

var defaultGLAccount = req.body.defaultGLAccount;

var nameToPrintOnCheckOr1099 = req.body.nameToPrintOnCheckOr1099;

var mailingName = req.body.mailingName;

var mailingStreet = req.body.mailingStreet;

var mailingCityStateZip = req.body.mailingCityStateZip;

var socialSecurityNumber = req.body.socialSecurityNumber;

var isNon1099Vendor = req.body.isNon1099Vendor;

var notes = req.body.notes;

var sql = `INSERT INTO vendorGeneralInformationC (name,isActive,street,zip,city,state,phone,phoneTwo,fax,contact,email,categoryId,accountNumber,defaultGLAccount,nameToPrintOnCheckOr1099,mailingName,mailingStreet,mailingCityStateZip,socialSecurityNumber,isNon1099Vendor,notes) VALUES ("${name}", "${isActive}", "${street}", "${zip}","${city}","${state}","${phone}","${phoneTwo}","${fax}","${contact}","${email}","${categoryId}","${accountNumber}","${defaultGLAccount}","${nameToPrintOnCheckOr1099}","${mailingName}","${mailingStreet}","${mailingCityStateZip}","${socialSecurityNumber}","${isNon1099Vendor}","${notes}")`;

dbConn.query(sql, function(err, result) {

if(err) {

res.status(500).send({ error: 'Something failed!' })

}

res.json({'status': 'success', id: result.insertId})

})

});

app.post('/createVendorBillHistory', function(req, res, next) {

var vendorId = req.body.vendorId;

var date = req.body.date;

var dateDue = req.body.dateDue;

var datePaid = req.body.datePaid;

var amount = req.body.amount;

var invoice = req.body.invoice;

var status = req.body.status;

var checkNumber = req.body.checkNumber;

var isCleared = req.body.isCleared;

var checkMemo = req.body.checkMemo;

//vendorId,date,dateDue,datePaid,amount,invoice,status,checkNumber,isCleared,checkMemo

var sql = `INSERT INTO addingVendor\_BillHistoryB (vendorId,date,dateDue,datePaid,amount,invoice,status,checkNumber,isCleared,checkMemo) VALUES ("${vendorId}", "${date}", "${dateDue}", "${datePaid}","${amount}","${invoice}","${status}","${checkNumber}","${isCleared}","${checkMemo}")`;

dbConn.query(sql, function(err, result) {

if(err) {

res.status(500).send({ error: 'Something failed!' })

}

res.json({'status': 'success', id: result.insertId})

})

});

app.post('/createAddingVehicleGeneralTen', function(req, res, next) {

//https://forums.mysql.com/read.php?51,506541,506541

//SELECT column\_name from information\_schema.columns where table\_schema = "dbOne" and table\_name = "addingVehicle\_General";

var mileage = req.body.mileage;

var isExempt = req.body.isExempt;

var locationCode = req.body.locationCode;

Var mileageIs = req.body.mileageIs

var newOrUseOrRebuilt = req.body.newOrUseOrRebuilt;

var isCertified = req.body.isCertified;

var vin = req.body.vin;

var year = req.body.year;

var transmission = req.body.transmission;

var transpeed = req.body.transpeed;

var driveTrain = req.body.driveTrain;

var make = req.body.make;

var model = req.body.model;

var style = req.body.style;

var bodyType = req.body.bodyType;

var exteriorColor = req.body.exteriorColor;

var exteriorColorTwo = req.body.exteriorColorTwo;

var colorDescription = req.body.colorDescription;

var interiorColor = req.body.interiorColor;

var tag = req.body.tag;

var decalNumber = req.body.decalNumber;

var county = req.body.county;

var titleApplication = req.body.titleApplication

var stateTitleIn = req.body.stateTitleIn;

var isTitleIn = req.body.isTitleIn;

var titleNumber = req.body.titleNumber;

//mileage,isExempt,locationCode,newOrUseOrRebuilt,isCertified,vin,year,transmission,transpeed,driveTrain,make,model,style,bodyType,exteriorColor,exteriorColorTwo,colorDescription,interiorColor,tag,decalNumber,county,titleApplication,stateTitleIn,isTitleIn,titleNumber,inspectionNumber,inspectionDate,inspectedBy,gpsSerialNumber,engineCylinders,engineSize,vehicleType,condition,weight,expirationDate,fuelType,mpgCombined,mpgCity,mpgHighway,ignitionKeyCode,doorKeyCode,valetKeyCode,isInspected,

//isDuplicateKeys,purchaseDate,originalCost,buyersFee,lotFee,addedCosts,floorPlanCost,totalCosts,vehicleSource,buyer,notes,isDraft,draftNumber,datePaid

var inspectionNumber = req.body.inspectionNumber;

var inspectionDate = req.body.inspectionDate;

var inspectedBy = req.body.inspectedBy;

var gpsSerialNumber = req.body.gpsSerialNumber;

var engineCylinders = req.body.engineCylinders;

var engineSize = req.body.engineSize;

var vehType = req.body.vehType;

var vehCondition = req.body.vehCondition;

var vehWeight = req.body.vehWeight;

var expirationDate = req.body.expirationDate;

var fuelType = req.body.fuelType;

var mpgCombined = req.body.mpgCombined;

var mpgCity = req.body.mpgCity;

var mpgHighway = req.body.mpgHighway;

var ignitionKeyCode = req.body.ignitionKeyCode;

var doorKeyCode = req.body.doorKeyCode;

var valetKeyCode = req.body.valetKeyCode;

var isInspected = req.body.isInspected;

var isDuplicateKeys = req.body.isDuplicateKeys;

var purchaseDate = req.body.purchaseDate;

var originalCost = req.body.originalCost;

var buyersFee = req.body.buyersFee;

var lotFee = req.body.lotFee;

var addedCosts = req.body.addedCosts;

var floorPlanCost = req.body.floorPlanCost;

var totalCosts = req.body.totalCosts;

var howDidYouPay = req.body.howDidYouPay;

var vehicleSource = req.body.vehicleSource;

var vendorId = req.body.vendorId;

var buyer = req.body.buyer;

var notes = req.body.notes;

var isDraft = req.body.isDraft;

var draftNumber = req.body.draftNumber;

var datePaid = req.body.datePaid;

var sql =`INSERT INTO addingVehicle\_GeneralTen (mileage,isExempt,locationCode,mileageIs,newOrUseOrRebuilt,isCertified,vin,year,transmission,transpeed,driveTrain,make,model,style,bodyType,exteriorColor,exteriorColorTwo,colorDescription,interiorColor,tag,decalNumber,county,titleApplication,stateTitleIn,isTitleIn,titleNumber,inspectionNumber,inspectionDate,inspectedBy,gpsSerialNumber,engineCylinders,engineSize,vehType,vehCondition,vehWeight,expirationDate,fuelType,mpgCombined,mpgCity,mpgHighway,ignitionKeyCode,doorKeyCode,valetKeyCode,isInspected,

isDuplicateKeys,purchaseDate,originalCost,buyersFee,lotFee,addedCosts,floorPlanCost,totalCosts,vehicleSource,buyer,notes,isDraft,draftNumber,datePaid) VALUES ("${mileage}", "${isExempt}", "${locationCode}","${mileageIs}", "${newOrUseOrRebuilt}", "${isCertified}", "${vin}", "${year}", "${transmission}", "${transpeed}", "${driveTrain}","${make}", "${model}", "${style}", "${bodyType}", "${exteriorColor}", "${exteriorColorTwo}", "${colorDescription}", "${interiorColor}", "${tag}", "${decalNumber}", "${county}", "${titleApplication}", "${stateTitleIn}", "${isTitleIn}","${titleNumber}", "${inspectionNumber}", "${inspectionDate}", "${inspectedBy}", "${gpsSerialNumber}", "${engineCylinders}", "${engineSize}", "${vehType}", "${vehCondition}", "${vehWeight}", "${expirationDate}", "${fuelType}", "${mpgCombined}", "${mpgCity}", "${mpgHighway}", "${ignitionKeyCode}", "${doorKeyCode}", "${valetKeyCode}", "${isInspected}", "${isDuplicateKeys}", "${purchaseDate}", "${originalCost}", "${buyersFee}", "${lotFee}", "${addedCosts}", "${floorPlanCost}", "${totalCosts}", "${vehicleSource}", "${buyer}", "${notes}", "${isDraft}", "${draftNumber}", "${datePaid}")`;

dbConn.query(sql, function(err, result) {

if(err) {

stockId = result.insertId;

res.status(500).send({ error: 'Something failed!',stockId: result.insertId})

}

res.json({'status': 'success', id: result.insertId,stockId: result.insertId})

// stockId = result.insertId;

})

//var date = req.body.date;

//var cost = req.body.cost;

//var vehicleId = req.body.vehicleId;

//var description = req.body.description;

//var ticket = req.body.ticket;

//var gl = req.body.gl;

/\*

//date,cost,vehicleId,description,ticket,gl

var description = '';

//const uuidv1 = require('uuid/v1');

//uuidv1(); // -> '6c84fb90-12c4-11e1-840d-7b25c5ee775a'

//var ticket = generator.next();

var id3 = intformat(id1, 'dec');

var ticket = id3;

let gl = 1;

//var gl = 1;

//stockId = 1001;

var stId = stockId;

var sqlA =`INSERT INTO addingVendor\_addedCosts (date,cost,vehicleId,description,ticket,gl) VALUES ("${purchaseDate}", "${addedCosts}", "${stId}", "${description}", "${ticket}", "${gl}")`;

dbConn.query(sqlA, function(err, result) {

if(err) {

res.status(500).send({ error: 'Something failed!',ticketno: ticket,stockId: stockId })

}

res.json({'status': 'success', id: result.insertId,ticketno: ticket,stockId: stockId });

// stockId = result.insertId;

})

\*/

});

// set port

app.listen(3000, function () {

console.log('Node app is running on port 3000');

});

module.exports = app;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

app.js

//https://w3path.com/node-js-crud-operation-with-express-and-mysql-database/

var http = require('http');

var express = require('express');

var app = express();

var bodyParser = require('body-parser');

var mysql = require('mysql');

var createError = require('http-errors');

//var express = require('express');

var path = require('path');

//var cookieParser = require('cookie-parser');

//var logger = require('morgan');

var expressValidator = require('express-validator');

var flash = require('express-flash');

var session = require('express-session');

//var bodyParser = require('body-parser');

app.use(session({

secret: '123456cat',

resave: false,

saveUninitialized: true,

cookie: { maxAge: 60000 }

}))

app.use(flash());

//app.use(expressValidator());

// connection configurations

var dbConn = mysql.createConnection({

host: 'localhost',

user: 'root',

password: 'krishna',

database: 'dbFive'

/\* host: 'localhost',

user: 'root',

password: '',

database: 'node\_js\_api'

\*/

});

// connect to database

dbConn.connect();

app.use(bodyParser.json());

app.use(bodyParser.urlencoded({

extended: false

}));

app.set('view engine', 'ejs');

const expressSession = require("express-session"),

cookieParser = require("cookie-parser"),

connectFlash = require("connect-flash");

app.use(cookieParser("secret\_passcode"));

app.use(expressSession({

secret: "secret\_passcode",

cookie: {

maxAge: 4000000

},

resave: false,

saveUninitialized: false

}));

//app.use(connectFlash());

app.use(flash());

//rest api to get billHistory

app.get('/vendorBillHistory', function (req, res) {

console.log(req);

dbConn.query('select \* from addingVendor\_BillHistoryB', function (error, results, fields) {

if (error) throw error;

res.render('vendor/billHistory',{page\_title:"addingVendor\_BillHistory - Node.js",data:results});

//res.end(JSON.stringify(results));

});

});

//rest api to get VendorAddedCosts

app.get('/VendorAddedCosts', function (req, res) {

console.log(req);

dbConn.query('select \* from addingVendor\_addedCosts', function (error, results, fields) {

if (error) throw error;

res.render('vendor/vendorAddedCosts',{page\_title:"VendorAddedCosts - Node.js",data:results});

//res.end(JSON.stringify(results));

});

});

//rest api to get VendorAddedCosts

app.get('/VendorAddedCosts', function (req, res) {

console.log(req);

dbConn.query('select \* from addingVendor\_addedCosts', function (error, results, fields) {

if (error) throw error;

res.render('vendor/vendorAddedCosts',{page\_title:"VendorAddedCosts - Node.js",data:results});

//res.end(JSON.stringify(results));

});

});

//rest api to get VendorAddedCosts

app.get('/VendorGLPostsOne', function (req, res) {

console.log(req);

dbConn.query('select \* from addingVendor\_GLPostsOne', function (error, results, fields) {

if (error) throw error;

res.render('vendor/vendorGLPostsOne',{page\_title:"addingVendorGLPostsOne - Node.js",data:results});

//res.end(JSON.stringify(results));

});

});

//rest api to get vendorGeneralInformation

app.get('/selectVendorGeneralInformation', function (req, res) {

console.log(req);

dbConn.query('select id, name, isActive, street, zip, city, state, phone, phoneTwo, fax, contact, email, categoryId, accountNumber, defaultGLAccount from vendorGeneralInformationC', function (error, results, fields) {

if (error) throw error;

res.render('vendor/selectVendorGeneralInformation',{page\_title:"selectVendorGeneralInformation - Node.js",data:results});

// res.end(JSON.stringify(results));

});

});

app.post('/createVendorBillHistory', function(req, res, next) {

var vendorId = req.body.vendorId;

var stockNumberId = req.body.stockNumberId;

var isAddToBill = req.body.isAddToBill;

var date = req.body.date;

var purchaseDate = req.body.purchaseDate;

var dateDue = req.body.dateDue;

//var datePaid = req.body.datePaid;

var datePaid = "";

var amount = req.body.amount;

var invoice = req.body.invoice;

var status = req.body.status;

var checkNumber = req.body.checkNumber;

//var isCleared = req.body.isCleared;

var isCleared = 0;

var checkMemo = req.body.checkMemo;

//vendorId,date,dateDue,datePaid,amount,invoice,status,checkNumber,isCleared,checkMemo

var sql = `INSERT INTO addingVendor\_BillHistoryB (vendorId,stockNumberId,isAddToBill,date,purchaseDate,dateDue,datePaid,amount,invoice,status,checkNumber,isCleared,checkMemo) VALUES ("${vendorId}", "${stockNumberId}", "${isAddToBill}", "${date}", "${purchaseDate}", "${dateDue}", "${datePaid}","${amount}","${invoice}","${status}","${checkNumber}","${isCleared}","${checkMemo}")`;

dbConn.query(sql, function(err, result) {

console.log(req.body);

if(err) {

// res.status(500).send({ error: 'Something failed!' })

}

else{

res.render('vendor/addingVehicle\_General', {

title: 'Form for addingVehicle\_General'

})

}

// res.json({'status': 'success', id: result.insertId})

})

});

//rest api to get vendorGeneralInformation

app.get('/vendorGeneralInformation', function (req, res) {

console.log(req);

dbConn.query('select \* from vendorGeneralInformationC', function (error, results, fields) {

if (error) throw error;

res.render('vendor/vendorGeneralInformationC',{page\_title:"vendorGeneralInformationC - Node.js",data:results});

// res.end(JSON.stringify(results));

});

});

// SHOW ADD USER FORM

app.get('/addingVendor\_General', function(req, res, next){

// render to views/user/add.ejs

res.render('vendor/addingVendor\_General', {

title: 'Form for addingVendor\_General'

})

})

app.get('/test', function(req, res, next){

// render to views/user/add.ejs

res.render('vendor/testOne', {

title: 'Form for addingVendor\_General'

})

})

app.get('/testTwo', function(req, res, next){

// render to views/user/add.ejs

res.render('vendor/testTwo', {

title: 'Form for addingVendor\_General'

})

})

// SHOW ADD USER FORM

app.get('/addingVehicle\_General', function(req, res, next){

// render to views/user/add.ejs

res.render('vendor/addingVehicle\_General', {

title: 'Form for addingVehicle\_General'

})

})

app.get('/addingVehicle', function(req, res, next){

// render to views/user/add.ejs

res.render('vendor/addingVehicle', {

title: 'Form for addingVehicle'

})

})

app.post('/save\_addingVendor\_General', function(req, res, next){

// req.check('name', 'Name is required').notEmpty() //Validate name

// req.check('email', 'A valid email is required').isEmail() //Validate email

//let name = req.body.name;

//let email = req.body.email;

//var errors = req.validationErrors()

console.log(req.body);

console.log('Input:: name='+req.body.name+' email='+ req.body.email);

// var sql = "insert into custName(name,email) values('"+ req.body.name +"','"+ req.body.email +"')";

// var sql = `INSERT INTO customerTwo (name, active, street, zip, city, state, phone, phone2, fax, contact, email, category, accoutNumber, defaultGLAccount, nameToPrintOnCheck, mailingName, mailingStreet, mailingCityStateZip, federalIdentificationOrSocialSecurityNumber) VALUES

// ("${name}", "${active}", "${street}", "${zip}","${city}","${state}","${phone}","${phone2}","${fax}","${contact}","${email}","${category}","${accoutNumber}","${defaultGLAccount}","${nameToPrintOnCheck}","${mailingName}","${mailingStreet}","${mailingCityStateZip}","${federalIdentificationOrSocialSecurityNumber}")`;

//var sql = 'insert into custName(name,email) values("${name}", "${email}")';

//var sql = `INSERT INTO custName (name, email) VALUES ("${name}", "${email}")`; undefined getting inserted

//var sql = `INSERT INTO custName (name, email) VALUES ("${name}", "${email}")`;

// var sql = 'insert into custName(name,email) values ('" + name + "', '" + email + "')'; SyntaxError: Unexpected string

// var sql = "insert into custName(name,email) values ('" + name + "', '" + email + "')";

//var sql = `INSERT INTO custName (name, email) VALUES ("${name}", "${email}")`;

var name = req.body.name;

var isActive = req.body.isActive;

var street = req.body.street;

var zip = req.body.zip;

var city = req.body.city;

var state = req.body.state;

var phone = req.body.phone;

var phoneTwo = req.body.phoneTwo;

var fax = req.body.fax;

//name,isActive,street,zip,city,state,phone,phoneTwo,fax,contact,email,categoryId,accoutNumber,defaultGLAccount,nameToPrintOnCheckOr1099,mailingName,mailingStreet,mailingCityStateZip,socialSecurityNumber,isNon1099Vendor,notes

var contact = req.body.contact;

var email = req.body.email;

var categoryId = req.body.categoryId;

var accountNumber = req.body.accoutNumber;

var defaultGLAccount = req.body.defaultGLAccount;

var nameToPrintOnCheckOr1099 = req.body.nameToPrintOnCheckOr1099;

var mailingName = req.body.mailingName;

var mailingStreet = req.body.mailingStreet;

var mailingCityStateZip = req.body.mailingCityStateZip;

var socialSecurityNumber = req.body.socialSecurityNumber;

var isNon1099Vendor = req.body.isNon1099Vendor;

var notes = req.body.notes;

var sql = `INSERT INTO vendorGeneralInformationC (name,isActive,street,zip,city,state,phone,phoneTwo,fax,contact,email,categoryId,accountNumber,defaultGLAccount,nameToPrintOnCheckOr1099,mailingName,mailingStreet,mailingCityStateZip,socialSecurityNumber,isNon1099Vendor,notes) VALUES ("${name}", "${isActive}", "${street}", "${zip}","${city}","${state}","${phone}","${phoneTwo}","${fax}","${contact}","${email}","${categoryId}","${accountNumber}","${defaultGLAccount}","${nameToPrintOnCheckOr1099}","${mailingName}","${mailingStreet}","${mailingCityStateZip}","${socialSecurityNumber}","${isNon1099Vendor}","${notes}")`;

console.log("Query :" + sql );

dbConn.query(sql, function (error, results, fields) {

console.log(JSON.stringify(results));

if(error){

req.flash("errorMessage", `${name}'s account not created!`);

res.locals.messageError = req.flash();

}else{

req.flash("successMessage", `${name}'s account created successfully!`);

res.locals.messageSuccess = req.flash();

res.render('vendor/addingVendor\_General', {

title: 'Form for addingVendor\_General'

})

// res.redirect('/addingVendor\_General');

// res.locals.messageSuccess = req.flash();

}

});

})

app.post('/createAddingVehicleGeneralTen', function(req, res, next) {

//https://forums.mysql.com/read.php?51,506541,506541

//SELECT column\_name from information\_schema.columns where table\_schema = "dbOne" and table\_name = "addingVehicle\_General";

var mileage = req.body.mileage;

var isExempt = req.body.isExempt;

var locationCode = req.body.locationCode;

var mileageIs = req.body.mileageIs;

var newOrUseOrRebuilt = req.body.newOrUseOrRebuilt;

var isCertified = req.body.isCertified;

var vin = req.body.vin;

var year = req.body.year;

var transmission = req.body.transmission;

var transpeed = req.body.transpeed;

var driveTrain = req.body.driveTrain;

var make = req.body.make;

var model = req.body.model;

var style = req.body.style;

var bodyType = req.body.bodyType;

var exteriorColor = req.body.exteriorColor;

var exteriorColorTwo = req.body.exteriorColorTwo;

var colorDescription = req.body.colorDescription;

var interiorColor = req.body.interiorColor;

var tag = req.body.tag;

var decalNumber = req.body.decalNumber;

var county = req.body.county;

var titleApplication = req.body.titleApplication

var stateTitleIn = req.body.stateTitleIn;

var isTitleIn = req.body.isTitleIn;

var titleNumber = req.body.titleNumber;

//mileage,isExempt,locationCode,newOrUseOrRebuilt,isCertified,vin,year,transmission,transpeed,driveTrain,make,model,style,bodyType,exteriorColor,exteriorColorTwo,colorDescription,interiorColor,tag,decalNumber,county,titleApplication,stateTitleIn,isTitleIn,titleNumber,inspectionNumber,inspectionDate,inspectedBy,gpsSerialNumber,engineCylinders,engineSize,vehicleType,condition,weight,expirationDate,fuelType,mpgCombined,mpgCity,mpgHighway,ignitionKeyCode,doorKeyCode,valetKeyCode,isInspected,

//isDuplicateKeys,purchaseDate,originalCost,buyersFee,lotFee,addedCosts,floorPlanCost,totalCosts,vehicleSource,buyer,notes,isDraft,draftNumber,datePaid

var inspectionNumber = req.body.inspectionNumber;

var inspectionDate = req.body.inspectionDate;

var inspectedBy = req.body.inspectedBy;

var gpsSerialNumber = req.body.gpsSerialNumber;

var engineCylinders = req.body.engineCylinders;

var engineSize = req.body.engineSize;

var vehType = req.body.vehType;

var vehCondition = req.body.vehCondition;

var vehWeight = req.body.vehWeight;

var expirationDate = req.body.expirationDate;

var fuelType = req.body.fuelType;

var mpgCombined = req.body.mpgCombined;

var mpgCity = req.body.mpgCity;

var mpgHighway = req.body.mpgHighway;

var ignitionKeyCode = req.body.ignitionKeyCode;

var doorKeyCode = req.body.doorKeyCode;

var valetKeyCode = req.body.valetKeyCode;

var isInspected = req.body.isInspected;

var isDuplicateKeys = req.body.isDuplicateKeys;

var purchaseDate = req.body.purchaseDate;

var originalCost = req.body.originalCost;

var buyersFee = req.body.buyersFee;

var lotFee = req.body.lotFee;

var addedCosts = req.body.addedCosts;

var floorPlanCost = req.body.floorPlanCost;

var totalCosts = req.body.totalCosts;

var howDidYouPay = req.body.howDidYouPay;

var vehicleSource = req.body.vehicleSource;

var vendorId = req.body.vendorId;

var buyer = req.body.buyer;

var notes = req.body.notes;

var isDraft = req.body.isDraft;

var draftNumber = req.body.draftNumber;

var datePaid = req.body.datePaid;

var sql =`INSERT INTO addingVehicle\_GeneralFourteen (mileage,isExempt,locationCode,mileageIs,newOrUseOrRebuilt,isCertified,vin,year,transmission,transpeed,driveTrain,make,model,style,bodyType,exteriorColor,exteriorColorTwo,colorDescription,interiorColor,tag,decalNumber,county,titleApplication,stateTitleIn,isTitleIn,titleNumber,inspectionNumber,inspectionDate,inspectedBy,gpsSerialNumber,engineCylinders,engineSize,vehType,vehCondition,vehWeight,expirationDate,fuelType,mpgCombined,mpgCity,mpgHighway,ignitionKeyCode,doorKeyCode,valetKeyCode,

isInspected,isDuplicateKeys,purchaseDate,originalCost,buyersFee,lotFee,addedCosts,floorPlanCost,totalCosts,howDidYouPay,vehicleSource,vendorId,buyer,notes,isDraft,draftNumber,datePaid) VALUES("${mileage}", "${isExempt}", "${locationCode}","${mileageIs}", "${newOrUseOrRebuilt}", "${isCertified}", "${vin}", "${year}", "${transmission}", "${transpeed}", "${driveTrain}","${make}", "${model}", "${style}", "${bodyType}", "${exteriorColor}", "${exteriorColorTwo}", "${colorDescription}", "${interiorColor}", "${tag}", "${decalNumber}", "${county}", "${titleApplication}", "${stateTitleIn}", "${isTitleIn}","${titleNumber}", "${inspectionNumber}", "${inspectionDate}", "${inspectedBy}", "${gpsSerialNumber}", "${engineCylinders}", "${engineSize}", "${vehType}", "${vehCondition}", "${vehWeight}", "${expirationDate}", "${fuelType}", "${mpgCombined}", "${mpgCity}", "${mpgHighway}","${ignitionKeyCode}", "${doorKeyCode}", "${valetKeyCode}", "${isInspected}", "${isDuplicateKeys}", "${purchaseDate}", "${originalCost}", "${buyersFee}", "${lotFee}", "${addedCosts}", "${floorPlanCost}", "${totalCosts}", "${howDidYouPay}", "${vehicleSource}", "${vendorId}", "${buyer}", "${notes}", "${isDraft}", "${draftNumber}", "${datePaid}")`;

console.log(req.body);

console.log("Query :" + sql );

dbConn.query(sql, function(err, results, fields) {

console.log(JSON.stringify(results));

if(err) {

// stockId = result.insertId;

req.flash("errorMessage", ` Account not created!`);

res.locals.messageError = req.flash();

// res.status(500).send({ error: 'Something failed!',stockId: result.insertId})

}

else{

// stockId = result.insertId;

req.flash("successMessage", `account created successfully!`);

res.locals.messageSuccess = req.flash();

res.render('vendor/addingVendor\_General', {

title: 'Form for addingVendor\_General'

})

}

// res.json({'status': 'success', id: result.insertId,stockId: result.insertId})

res.end(JSON.stringify(results));

// res.json({'status': 'success', id: result.insertId,stockId: result.insertId})

// stockId = result.insertId;

});

})

app.post('/createAddingVehicleGeneralFourteen', function(req, res, next) {

var mileage = req.body.mileage;

var isExempt = req.body.isExempt;

var locationCode = req.body.locationCode;

var mileageIs = req.body.mileageIs;

var newOrUseOrRebuilt = req.body.newOrUseOrRebuilt;

var isCertified = req.body.isCertified;

var vin = req.body.vin;

var year = req.body.year;

var transmission = req.body.transmission;

var transpeed = req.body.transpeed;

var driveTrain = req.body.driveTrain;

var make = req.body.make;

var model = req.body.model;

var style = req.body.style;

var bodyType = req.body.bodyType;

var exteriorColor = req.body.exteriorColor;

var exteriorColorTwo = req.body.exteriorColorTwo;

var colorDescription = req.body.colorDescription;

var interiorColor = req.body.interiorColor;

var tag = req.body.tag;

var decalNumber = req.body.decalNumber;

var county = req.body.county;

var titleApplication = req.body.titleApplication

var stateTitleIn = req.body.stateTitleIn;

var isTitleIn = req.body.isTitleIn;

var titleNumber = req.body.titleNumber;

//mileage,isExempt,locationCode,newOrUseOrRebuilt,isCertified,vin,year,transmission,transpeed,driveTrain,make,model,style,bodyType,exteriorColor,exteriorColorTwo,colorDescription,interiorColor,tag,decalNumber,county,titleApplication,stateTitleIn,isTitleIn,titleNumber,inspectionNumber,inspectionDate,inspectedBy,gpsSerialNumber,engineCylinders,engineSize,vehicleType,condition,weight,expirationDate,fuelType,mpgCombined,mpgCity,mpgHighway,ignitionKeyCode,doorKeyCode,valetKeyCode,isInspected,

//isDuplicateKeys,purchaseDate,originalCost,buyersFee,lotFee,addedCosts,floorPlanCost,totalCosts,vehicleSource,buyer,notes,isDraft,draftNumber,datePaid

var inspectionNumber = req.body.inspectionNumber;

var inspectionDate = req.body.inspectionDate;

var inspectedBy = req.body.inspectedBy;

var gpsSerialNumber = req.body.gpsSerialNumber;

var engineCylinders = req.body.engineCylinders;

var engineSize = req.body.engineSize;

var vehType = req.body.vehType;

var vehCondition = req.body.vehCondition;

var vehWeight = req.body.vehWeight;

var expirationDate = req.body.expirationDate;

var fuelType = req.body.fuelType;

var mpgCombined = req.body.mpgCombined;

var mpgCity = req.body.mpgCity;

var mpgHighway = req.body.mpgHighway;

var ignitionKeyCode = req.body.ignitionKeyCode;

var doorKeyCode = req.body.doorKeyCode;

var valetKeyCode = req.body.valetKeyCode;

var isInspected = req.body.isInspected;

var isDuplicateKeys = req.body.isDuplicateKeys;

var purchaseDate = req.body.purchaseDate;

var originalCost = req.body.originalCost;

var buyersFee = req.body.buyersFee;

var lotFee = req.body.lotFee;

var addedCosts = req.body.addedCosts;

var floorPlanCost = req.body.floorPlanCost;

var totalCosts = req.body.totalCosts;

var howDidYouPay = req.body.howDidYouPay;

var vehicleSource = req.body.vehicleSource;

var vendorId = req.body.vendorId;

var buyer = req.body.buyer;

var notes = req.body.notes;

var isDraft = req.body.isDraft;

var draftNumber = req.body.draftNumber;

var datePaid = req.body.datePaid;

var sql =`INSERT INTO addingVehicle\_GeneralFourteen (mileage,isExempt,locationCode,mileageIs,newOrUseOrRebuilt,isCertified,vin,year,transmission,transpeed,driveTrain,make,model,style,bodyType,exteriorColor,exteriorColorTwo,colorDescription,interiorColor,tag,decalNumber,county,titleApplication,stateTitleIn,isTitleIn,titleNumber,inspectionNumber,inspectionDate,inspectedBy,gpsSerialNumber,engineCylinders,engineSize,vehType,vehCondition,vehWeight,expirationDate,fuelType,mpgCombined,mpgCity,mpgHighway,ignitionKeyCode,doorKeyCode,valetKeyCode,isInspected,isDuplicateKeys,purchaseDate,originalCost,buyersFee,lotFee,addedCosts,floorPlanCost,totalCosts,howDidYouPay,vehicleSource,vendorId,buyer,notes,isDraft,draftNumber,datePaid) VALUES ("${mileage}", "${isExempt}", "${locationCode}","${mileageIs}", "${newOrUseOrRebuilt}", "${isCertified}", "${vin}", "${year}", "${transmission}", "${transpeed}", "${driveTrain}","${make}", "${model}", "${style}", "${bodyType}", "${exteriorColor}", "${exteriorColorTwo}", "${colorDescription}", "${interiorColor}", "${tag}", "${decalNumber}", "${county}", "${titleApplication}", "${stateTitleIn}", "${isTitleIn}","${titleNumber}", "${inspectionNumber}", "${inspectionDate}", "${inspectedBy}", "${gpsSerialNumber}", "${engineCylinders}", "${engineSize}", "${vehType}", "${vehCondition}", "${vehWeight}", "${expirationDate}", "${fuelType}", "${mpgCombined}", "${mpgCity}", "${mpgHighway}","${ignitionKeyCode}", "${doorKeyCode}", "${valetKeyCode}", "${isInspected}", "${isDuplicateKeys}", "${purchaseDate}", "${originalCost}", "${buyersFee}", "${lotFee}", "${addedCosts}", "${floorPlanCost}", "${totalCosts}", "${howDidYouPay}", "${vehicleSource}", "${vendorId}", "${buyer}", "${notes}", "${isDraft}", "${draftNumber}", "${datePaid}")`;

dbConn.query(sql, function(err, result) {

console.log(req.body);

console.log("Query :" + sql );

console.log(JSON.stringify(result));

if(err) {

// stockId = result.insertId;

// res.status(500).send({ error: 'Something failed!',stockId: result.insertId})

}

else{

req.flash("successMessage", `Vehicle created successfully!`);

res.locals.messageSuccess = req.flash();

res.render('vendor/addingVehicle\_General', {

title: 'Form for addingVehicle\_General'

})

}

// res.json({'status': 'success', id: result.insertId,stockId: result.insertId})

// res.end(JSON.stringify(results));

// res.json({'status': 'success', id: result.insertId,stockId: result.insertId})

// stockId = result.insertId;

})

})

app.post('/createAddingVehicleGeneralFifteen', function(req, res, next) {

var mileage = req.body.mileage;

var isExempt = req.body.isExempt;

var locationCode = req.body.locationCode;

var mileageIs = req.body.mileageIs;

var newOrUseOrRebuilt = req.body.newOrUseOrRebuilt;

var isCertified = req.body.isCertified;

var vin = req.body.vin;

var year = req.body.year;

var transmission = req.body.transmission;

var make = req.body.make;

var model = req.body.model;

var style = req.body.style;

var bodyType = req.body.bodyType;

var exteriorColor = req.body.exteriorColor;

var exteriorColorTwo = req.body.exteriorColorTwo;

var colorDescription = req.body.colorDescription;

var interiorColor = req.body.interiorColor;

var tag = req.body.tag;

var decalNumber = req.body.decalNumber;

var county = req.body.county;

var titleApplication = req.body.titleApplication

var stateTitleIn = req.body.stateTitleIn;

var isTitleIn = req.body.isTitleIn;

var titleNumber = req.body.titleNumber;

//mileage,isExempt,locationCode,newOrUseOrRebuilt,isCertified,vin,year,transmission,transpeed,driveTrain,make,model,style,bodyType,exteriorColor,exteriorColorTwo,colorDescription,interiorColor,tag,decalNumber,county,titleApplication,stateTitleIn,isTitleIn,titleNumber,inspectionNumber,inspectionDate,inspectedBy,gpsSerialNumber,engineCylinders,engineSize,vehicleType,condition,weight,expirationDate,fuelType,mpgCombined,mpgCity,mpgHighway,ignitionKeyCode,doorKeyCode,valetKeyCode,isInspected,

//isDuplicateKeys,purchaseDate,originalCost,buyersFee,lotFee,addedCosts,floorPlanCost,totalCosts,vehicleSource,buyer,notes,isDraft,draftNumber,datePaid

var inspectionNumber = req.body.inspectionNumber;

var inspectionDate = req.body.inspectionDate;

var isInspected = req.body.isInspected;

var isDuplicateKeys = req.body.isDuplicateKeys;

var inspectedBy = req.body.inspectedBy;

var gpsSerialNumber = req.body.gpsSerialNumber;

var sql =`INSERT INTO addingVehicle\_GeneralFifteen (mileage,isExempt,locationCode,mileageIs,newOrUseOrRebuilt,isCertified,vin,year,transmission,make,model,style,bodyType,exteriorColor,exteriorColorTwo,colorDescription,interiorColor,tag,decalNumber,county,titleApplication,stateTitleIn,isTitleIn,titleNumber,inspectionNumber,inspectionDate,isInspected,isDuplicateKeys,inspectedBy,gpsSerialNumber) VALUES("${mileage}", "${isExempt}", "${locationCode}","${mileageIs}", "${newOrUseOrRebuilt}", "${isCertified}", "${vin}", "${year}", "${transmission}","${make}", "${model}", "${style}", "${bodyType}", "${exteriorColor}", "${exteriorColorTwo}", "${colorDescription}", "${interiorColor}", "${tag}", "${decalNumber}", "${county}", "${titleApplication}", "${stateTitleIn}", "${isTitleIn}","${titleNumber}", "${inspectionNumber}", "${inspectionDate}","${isInspected}", "${isDuplicateKeys}", "${inspectedBy}", "${gpsSerialNumber}")`;

dbConn.query(sql, function(err, result) {

console.log(req.body);

console.log("Query :" + sql );

console.log(JSON.stringify(result));

if(err) {

// stockId = result.insertId;

// res.status(500).send({ error: 'Something failed!',stockId: result.insertId})

}

// res.json({'status': 'success', id: result.insertId,stockId: result.insertId})

res.render('vendor/addingVendor\_General', {

title: 'Form for addingVendor\_General'

})

// res.end(JSON.stringify(results));

// res.json({'status': 'success', id: result.insertId,stockId: result.insertId})

// stockId = result.insertId;

})

})

app.post('/createAddingVehicleGeneralSixteen', function(req, res, next) {

var transpeed = req.body.transpeed;

var driveTrain = req.body.driveTrain;

var engineCylinders = req.body.engineCylinders;

var engineSize = req.body.engineSize;

var vehType = req.body.vehType;

var vehCondition = req.body.vehCondition;

var vehWeight = req.body.vehWeight;

var expirationDate = req.body.expirationDate;

var fuelType = req.body.fuelType;

var mpgCombined = req.body.mpgCombined;

var mpgCity = req.body.mpgCity;

var mpgHighway = req.body.mpgHighway;

var ignitionKeyCode = req.body.ignitionKeyCode;

var doorKeyCode = req.body.doorKeyCode;

var valetKeyCode = req.body.valetKeyCode;

var isInspected = req.body.isInspected;

var isDuplicateKeys = req.body.isDuplicateKeys;

var purchaseDate = req.body.purchaseDate;

var originalCost = req.body.originalCost;

var buyersFee = req.body.buyersFee;

var lotFee = req.body.lotFee;

var addedCosts = req.body.addedCosts;

var floorPlanCost = req.body.floorPlanCost;

var totalCosts = req.body.totalCosts;

var howDidYouPay = req.body.howDidYouPay;

var vehicleSource = req.body.vehicleSource;

var vendorId = req.body.vendorId;

var buyer = req.body.buyer;

var notes = req.body.notes;

var isDraft = req.body.isDraft;

var draftNumber = req.body.draftNumber;

var datePaid = req.body.datePaid;

var sql =`INSERT INTO addingVehicle\_GeneralSixteen (transpeed,driveTrain,engineCylinders,engineSize,vehType,vehCondition,vehWeight,expirationDate,fuelType,mpgCombined,mpgCity,mpgHighway,ignitionKeyCode,doorKeyCode,valetKeyCode,purchaseDate,originalCost,buyersFee,lotFee,addedCosts,floorPlanCost,totalCosts,howDidYouPay,vehicleSource,vendorId,buyer,notes,isDraft,draftNumber,datePaid) VALUES( "${transpeed}", "${driveTrain}", "${engineCylinders}", "${engineSize}", "${vehType}", "${vehCondition}", "${vehWeight}", "${expirationDate}", "${fuelType}", "${mpgCombined}", "${mpgCity}", "${mpgHighway}","${ignitionKeyCode}", "${doorKeyCode}", "${valetKeyCode}", "${purchaseDate}", "${originalCost}", "${buyersFee}", "${lotFee}", "${addedCosts}", "${floorPlanCost}", "${totalCosts}", "${howDidYouPay}", "${vehicleSource}", "${vendorId}", "${buyer}", "${notes}", "${isDraft}", "${draftNumber}", "${datePaid}")`;

dbConn.query(sql, function(err, result) {

console.log(req.body);

console.log("Query :" + sql );

console.log(JSON.stringify(result));

if(err) {

// stockId = result.insertId;

// res.status(500).send({ error: 'Something failed!',stockId: result.insertId})

}

res.render('vendor/addingVendor\_General', {

title: 'Form for addingVendor\_General'

})

// res.json({'status': 'success', id: result.insertId,stockId: result.insertId})

// res.end(JSON.stringify(results));

// res.json({'status': 'success', id: result.insertId,stockId: result.insertId})

// stockId = result.insertId;

})

})

// set port

app.listen(3000, function () {

console.log('Node app is running on port 3000');

});

module.exports = app;