Josh Mkhari

Student number: 20104681

PATHWAY: CLDV6211

Lecturer: Isaac Leshaba

7 July 2021

Cloud Development Task 3

TABLE OF CONTENTS

TABLES. DIAGRAMS AND FIGURES

1. INTRODUCTION

Creators of databases face many challenges in the development of their databases as such security is a top concern that they all need to look out for. SQL queries can be made in such a way that the database creator can do their best to secure it. (Tunggal, 2020).

Web app LINK : https://domingoroofworksnj.azurewebsites.net

2. SQL Query Code Stored Procedures

```
-- Customer SUFF
CREATE PROCEDURE SP_GetAllCustomer
AS
BEGIN
SELECT * FROM CUSTOMER
END



CREATE PROCEDURE SP_GetCustomer
(
@CUSTOMER_ID INT  = 0
)
AS
BEGIN
SELECT * FROM CUSTOMER WHERE CUSTOMER_ID = @CUSTOMER_ID
END

CREATE PROCEDURE SP_GetCurrentCustomer
AS
BEGIN
Select TOP 1 CUSTOMER_ID FROM CUSTOMER
ORDER BY CUSTOMER_ID DESC
```

```
END


CREATE PROCEDURE SP_GetCurrentJobCardNo
AS
BEGIN
Select TOP 1 JOB_CARD_NO FROM job
ORDER BY JOB_ID DESC
END

CREATE PROCEDURE SP_UpdateCustomer
(
@CUSTOMER_ID INT = 0,
@NAME VARCHAR (50) = '',
@SURNAME VARCHAR (50)  = '',
@ADDRESS_LINE_ONE VARCHAR (250) = '',
@ADDRESS_LINE_TWO VARCHAR (250) = '',
@CITY VARCHAR (100) = '',
@POSTAL_CODE VARCHAR (5) = ''
)
AS
BEGIN
UPDATE CUSTOMER
SET [NAME] = @NAME,
SURNAME = @SURNAME,
ADDRESS_LINE_ONE = @ADDRESS_LINE_ONE,
ADDRESS_LINE_TWO = @ADDRESS_LINE_TWO,
CITY = @CITY,
POSTAL_CODE = @POSTAL_CODE
WHERE CUSTOMER_ID = @CUSTOMER_ID
END
```

```
CREATE PROCEDURE SP_InsertCustomer
(
@NAME VARCHAR (50) = '',
@SURNAME VARCHAR (50)  = '',
@ADDRESS_LINE_ONE VARCHAR (250) = '',
@ADDRESS_LINE_TWO VARCHAR (250) = '',
@CITY VARCHAR (100) = '',
@POSTAL_CODE VARCHAR (5) = ''
)
AS
BEGIN
INSERT INTO CUSTOMER([NAME],SURNAME,ADDRESS_LINE_ONE,ADDRESS_LINE_TWO,CITY,POSTAL_CODE)
VALUES (@NAME,@SURNAME,@ADDRESS_LINE_ONE,@ADDRESS_LINE_TWO,@CITY,@POSTAL_CODE)
END


CREATE PROCEDURE SP_DeleteCustomer
(
@CUSTOMER_ID INT  = 0
)
AS
BEGIN
DELETE FROM QUOTATION WHERE CUSTOMER_ID = @CUSTOMER_ID
DELETE FROM CUSTOMER WHERE CUSTOMER_ID = @CUSTOMER_ID
END


CREATE PROCEDURE SP_CreateJob
(
@JOB_CARD_NO INT   = 0,
@NO_OF_DAYS INT = 0,
@CUSTOMER_ID INT = 0
)
AS
BEGIN
INSERT INTO JOB(JOB_CARD_NO,NO_OF_DAYS)
VALUES (@JOB_CARD_NO,@NO_OF_DAYS)
INSERT INTO QUOTATION(JOB_CARD_NO,CUSTOMER_ID)
VALUES (@JOB_CARD_NO,@CUSTOMER_ID)
END

CREATE PROCEDURE SP_AddEqID
(
@RateID int = 0,
@StandardFloorBoards int =0
)
AS
BEGIN
INSERT INTO EQUIPTMENT_MATERIALS(RATE_ID,STANDARD_FLOOR_BOARDS)
VALUES (@RateID,@StandardFloorBoards)
END

CREATE PROCEDURE SP_GetCurrentEquip
AS
BEGIN
Select TOP 1 EQUIPTMENT_MATERIALS_ID FROM EQUIPTMENT_MATERIALS
ORDER BY EQUIPTMENT_MATERIALS_ID DESC
END

CREATE PROCEDURE SP_AddJobEMaterials
(
@EquiptmentMaterialsID int = 0,
@JobCardNo int = 0
```

```
)
AS
BEGIN
INSERT INTO JOB_EQUIPTMENT_MATERIALS(EQUIPTMENT_MATERIALS_ID,JOB_CARD_NO)
VALUES (@EquiptmentMaterialsID,@JobCardNo)
END

CREATE PROCEDURE SP_SelectMaterials
(
@EQUIPTMENT_MATERIALS_ID INT = 0,
@POWER_POINTS INT = 0,
@STANDARD_ELECTRICAL_WIRING INT = 0
)
AS
BEGIN
INSERT INTO MATERIALS(EQUIPTMENT_MATERIALS_ID,POWER_POINTS,STANDARD_ELECTRICAL_WIRING)
VALUES(@EQUIPTMENT_MATERIALS_ID,@POWER_POINTS,@STANDARD_ELECTRICAL_WIRING)

END
------------------------------------------------------------------
--Employee Stuff
CREATE PROCEDURE SP_GetAllEmployee
AS
BEGIN
SELECT * FROM EMPLOYEE
END

CREATE PROCEDURE SP_InsertEmployee
(
@EMPLOYEE_NO VARCHAR (6)  = '',
@NAME VARCHAR (50) = '',
@SURNAME VARCHAR (50)  = ''
)
AS
BEGIN
INSERT INTO EMPLOYEE (EMPLOYEE_NO,[NAME],SURNAME)
VALUES (@EMPLOYEE_NO,@NAME,@SURNAME)
END

CREATE PROCEDURE SP_GetEmployee
(
@EMPLOYEE_NO VARCHAR (6)  = ''
)
AS
BEGIN
SELECT * FROM EMPLOYEE WHERE EMPLOYEE_NO = @EMPLOYEE_NO
END

CREATE PROCEDURE SP_UpdateEmployee
(
@EMPLOYEE_NO VARCHAR (6)  = '',
@NAME VARCHAR (50) = '',
@SURNAME VARCHAR (50)  = ''
)
AS
BEGIN
UPDATE EMPLOYEE
SET [NAME] = @NAME,
SURNAME = @SURNAME
WHERE EMPLOYEE_NO = @EMPLOYEE_NO
END
```

```
----------------------------------------------------------------
--JobCard Stuff
CREATE PROCEDURE SP_GetAllJobCards
AS
BEGIN
select j.JOB_CARD_NO as [JOB_CARD_NO],j.NO_OF_DAYS as [NO_OF_DAYS], COUNT(ejc.EMPLOYEE_NO) as [Employees Assigned] from job j
left join EMPLOYEE_JOB_CARD ejc on ejc.JOB_CARD_NO = j.JOB_CARD_NO
group by j.JOB_ID,j.JOB_CARD_NO,j.NO_OF_DAYS
order by COUNT(ejc.EMPLOYEE_NO) ASC
END

CREATE PROCEDURE SP_EmployeesWorkingOnJob
(
@JobCardNo int = 0
)
AS
BEGIN
Select  ej.EMPLOYEE_NO as EmpNo, e.NAME as [Employee Name], e.SURNAME as [Employee Surname] from EMPLOYEE_JOB_CARD ej
inner join EMPLOYEE e on e.EMPLOYEE_NO = ej.EMPLOYEE_NO
where JOB_CARD_NO = @JobCardNo
END

CREATE PROCEDURE SP_GetRateForJobCard
(
@JobCardNo int = 0
)
AS
BEGIN
Select  em.RATE_ID as [JobType] from EQUIPTMENT_MATERIALS em
inner join JOB_EQUIPTMENT_MATERIALS jem on jem.EQUIPTMENT_MATERIALS_ID = em.EQUIPTMENT_MATERIALS_ID
```

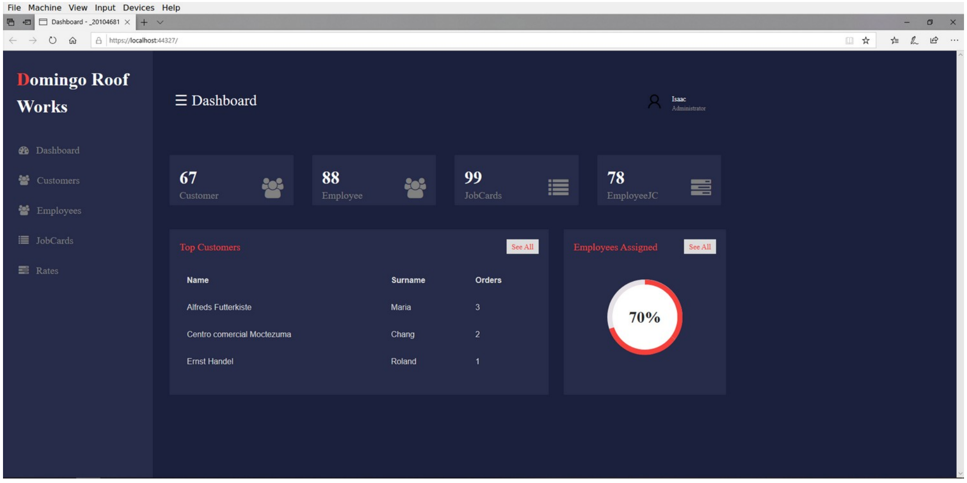
```
where JOB_CARD_NO = @JobCardNo
END

CREATE PROCEDURE SP_AddEmployeeToJobCard
(
@EmployeeNo VARCHAR(6) ='',
@JobCard int = 0
)
AS
BEGIN
INSERT INTO EMPLOYEE_JOB_CARD(EMPLOYEE_NO,JOB_CARD_NO)
VALUES (@EmployeeNo,@JobCard)
END

create PROCEDURE SP_GetInvoiceFloorBoarding
(
@JobCardNo int = 0
)
AS
BEGIN
select j.JOB_CARD_NO as [JOB_CARD_NO],j.NO_OF_DAYS as
[NO_OF_DAYS],e.EMPLOYEE_NO as [EmpNo], e.NAME as [Employee
Name],e.SURNAME as [Employee Surname],c.NAME as [Customer
Name],c.SURNAME as [Customer Surname],c.ADDRESS_LINE_ONE as [Add
One],c.ADDRESS_LINE_TWO as [Add two],c.CITY as [city],c.POSTAL_CODE as
[code], r.JOB_TYPE as [Job Type],eq.STANDARD_FLOOR_BOARDS as
[FloorBoards],'R' + CAST(r.RATE AS VARCHAR(15)) as [Rate], 'R' + CAST(r.RATE *
j.NO_OF_DAYS AS VARCHAR(15)) AS Subtotal, 'R' + CAST(0.14*r.RATE *
j.NO_OF_DAYS AS VARCHAR(15)) AS [VAT], 'R' + CAST(((0.14*r.RATE *
j.NO_OF_DAYS) + (RATE * j.NO_OF_DAYS)) AS  VARCHAR(15)) AS [Total:]  from
job j
left join EMPLOYEE_JOB_CARD ejc on ejc.JOB_CARD_NO = j.JOB_CARD_NO
left join EMPLOYEE e on ejc.EMPLOYEE_NO = e.EMPLOYEE_NO
inner join QUOTATION q on q.JOB_CARD_NO = j.JOB_CARD_NO
inner join CUSTOMER c on c.CUSTOMER_ID = q.CUSTOMER_ID
inner join JOB_EQUIPTMENT_MATERIALS jec on jec.JOB_CARD_NO =
j.JOB_CARD_NO

inner join EQUIPTMENT_MATERIALS eq on eq.EQUIPTMENT_MATERIALS_ID =
jec.EQUIPTMENT_MATERIALS_ID

inner join RATE r on r.RATE_ID = eq.RATE_ID

where j.JOB_CARD_NO = @JobCardNo

END


CREATE PROCEDURE SP_GetInvoiceSemi

(

@JobCardNo int = 0

)

AS

BEGIN

select j.JOB_CARD_NO as [JOB_CARD_NO],j.NO_OF_DAYS as
[NO_OF_DAYS],e.EMPLOYEE_NO as [EmpNo], e.NAME as [Employee
Name],e.SURNAME as [Employee Surname],c.NAME as [Customer
Name],c.SURNAME as [Customer Surname],c.ADDRESS_LINE_ONE as [Add
One],c.ADDRESS_LINE_TWO as [Add two],c.CITY as [city],c.POSTAL_CODE as
[code], r.JOB_TYPE as [Job Type],eq.STANDARD_FLOOR_BOARDS as
[FloorBoards],m.POWER_POINTS as
[PowerPoints],m.STANDARD_ELECTRICAL_WIRING as [SEW],'R' + CAST(r.RATE AS
VARCHAR(15)) as [Rate], 'R' + CAST(r.RATE * j.NO_OF_DAYS AS VARCHAR(15))
AS Subtotal, 'R' + CAST(0.14*r.RATE * j.NO_OF_DAYS AS VARCHAR(15)) AS [VAT],
'R' + CAST(((0.14*r.RATE * j.NO_OF_DAYS) + (RATE * j.NO_OF_DAYS)) AS
VARCHAR(15)) AS [Total:]  from job j

left join EMPLOYEE_JOB_CARD ejc on ejc.JOB_CARD_NO = j.JOB_CARD_NO

left join EMPLOYEE e on ejc.EMPLOYEE_NO = e.EMPLOYEE_NO

inner join QUOTATION q on q.JOB_CARD_NO = j.JOB_CARD_NO

inner join CUSTOMER c on c.CUSTOMER_ID = q.CUSTOMER_ID

inner join JOB_EQUIPTMENT_MATERIALS jec on jec.JOB_CARD_NO =
j.JOB_CARD_NO

inner join EQUIPTMENT_MATERIALS eq on eq.EQUIPTMENT_MATERIALS_ID =
jec.EQUIPTMENT_MATERIALS_ID

inner join RATE r on r.RATE_ID = eq.RATE_ID

inner join MATERIALS m on m.EQUIPTMENT_MATERIALS_ID =
eq.EQUIPTMENT_MATERIALS_ID

where j.JOB_CARD_NO = @JobCardNo
```

```
END

CREATE PROCEDURE SP_DeleteEquiptmentJob
(
@JobCardNo int = 0
)
AS
BEGIN
DELETE FROM JOB_EQUIPTMENT_MATERIALS WHERE JOB_CARD_NO = 1017
END



CREATE PROCEDURE SP_DeleteEmployeeJob
(
@JobCardNo int = 0
)
AS
BEGIN
DELETE FROM EMPLOYEE_JOB_CARD WHERE JOB_CARD_NO = @JobCardNo
END

create PROCEDURE SP_DeleteJobCard
(
@JobCardNo int = 0
)
AS
BEGIN
DELETE FROM QUOTATION WHERE JOB_CARD_NO = @JobCardNo
DELETE FROM job WHERE JOB_CARD_NO = @JobCardNo
END

create PROCEDURE SP_CheckJobCard
```

```
(
@JobCardNo int = 0
)
AS
BEGIN
SELECT j.JOB_CARD_NO as [JOB_CARD_NO],JEM.EQUIPTMENT_MATERIALS_ID as [EquipID] FROM job j
LEFT JOIN EMPLOYEE_JOB_CARD ej on ej.JOB_CARD_NO = j.JOB_CARD_NO
LEFT JOIN JOB_EQUIPTMENT_MATERIALS jem on jem.JOB_CARD_NO = j.JOB_CARD_NO
WHERE j.JOB_CARD_NO = 1017
END


------------------------------------------------------------------
--Rate Stuff
CREATE PROCEDURE SP_GetRates
AS
BEGIN
SELECT RATE,JOB_TYPE FROM RATE
END


CREATE PROCEDURE SP_GetRate
(
@JobType Varchar(50) = ''
)
AS
BEGIN
SELECT RATE FROM RATE
where JOB_TYPE = @JobType
END


CREATE PROCEDURE SP_UpdateRate
(
@JobType Varchar(50) = '',
@Rate int= 0
)
AS
BEGIN
UPDATE RATE
SET RATE = @Rate
WHERE JOB_TYPE = @JobType
END
```

3. How to Navigate the web app

Q 1.1)

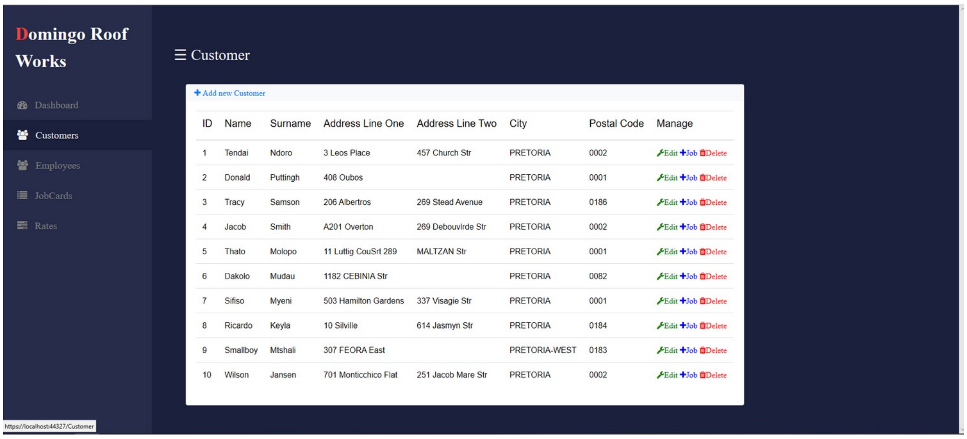


Top Customers See All

Name	Surname	Orders
Alfreds Futterkiste	Maria	3
Centro comercial Moctezuma	Chang	2
Ernst Handel	Roland	1

(MBuschi, 2021) (Stevestein, 2019) both sources helped me identify the ASPNETCORE_ENVIRONMENT error I ran into which was preventing me from being able to access my published web app.

The dashboard is the first page you are met with and it has a bunch of objects with data. To navigate simply use the navbar to the left.



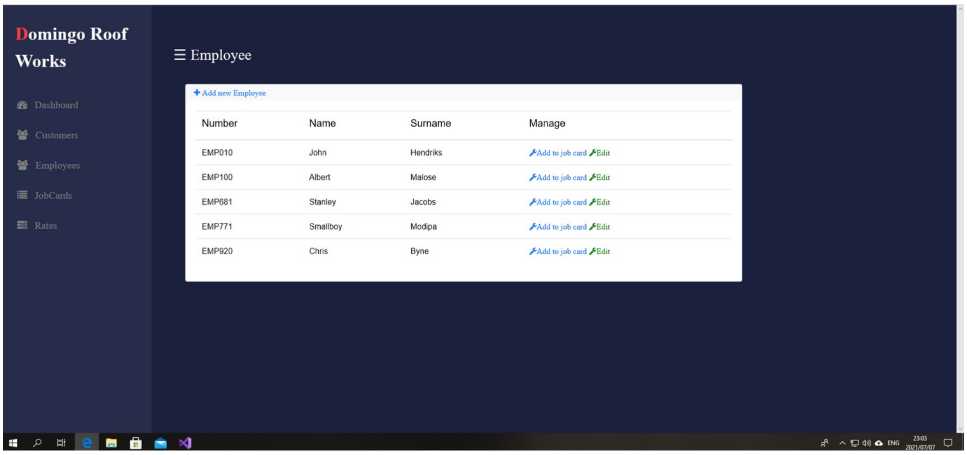
☰ Customer

+ Add new Customer

ID	Name	Surname	Address Line One	Address Line Two	City	Postal Code	Manage
1	Tendai	Ndoro	3 Leos Place	457 Church Str	PRETORIA	0002	Edit +Job Delete
2	Donald	Puttingh	408 Oubos		PRETORIA	0001	Edit +Job Delete
3	Tracy	Samson	206 Albertros	269 Stead Avenue	PRETORIA	0186	Edit +Job Delete
4	Jacob	Smith	A201 Overton	269 Debouvlrde Str	PRETORIA	0002	Edit +Job Delete
5	Thato	Molopo	11 Luttig CouSrt 289	MALTZAN Str	PRETORIA	0001	Edit +Job Delete
6	Dakolo	Mudau	1182 CEBINIA Str		PRETORIA	0082	Edit +Job Delete
7	Sifiso	Myeni	503 Hamilton Gardens	337 Visagie Str	PRETORIA	0001	Edit +Job Delete
8	Ricardo	Keyla	10 Silville	614 Jasmyn Str	PRETORIA	0184	Edit +Job Delete
9	Smallboy	Mtshali	307 FEORA East		PRETORIA-WEST	0183	Edit +Job Delete
10	Wilson	Jansen	701 Monticchico Flat	251 Jacob Mare Str	PRETORIA	0002	Edit +Job Delete

To login I had to whitelist my IP address (VanMSFT, 2020).

The Customer page offers the option to edit the customer, add a job for the customer(adding a job card), adding a new customer and deleting a customer.



☰ Employee

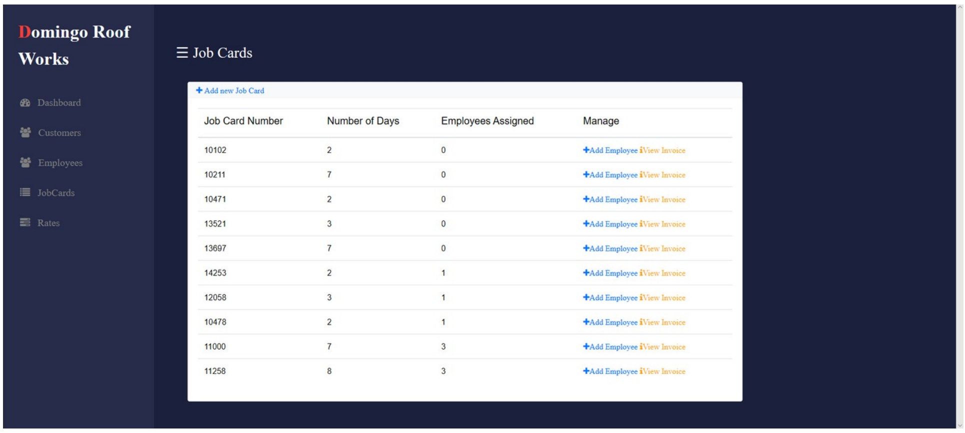
+ Add new Employee

Number	Name	Surname	Manage
EMP010	John	Hendriks	Add to job card Edit
EMP100	Albert	Malose	Add to job card Edit
EMP681	Stanley	Jacobs	Add to job card Edit
EMP771	Smallboy	Modipa	Add to job card Edit
EMP920	Chris	Byne	Add to job card Edit

The employee page offers the option to add an employee to a job card and edit and adding a new employee

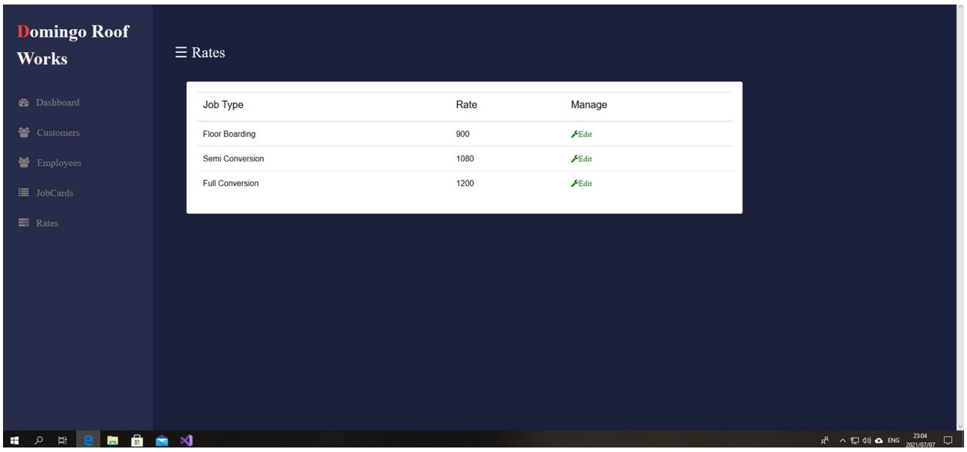
Using the query editor (Erkec, 2018) I was able to create my database on Azure as the Storage accounts were not allowing imports.

The job card page allows you to add an employee to a job card or view invoices for each job card showing the relevant information



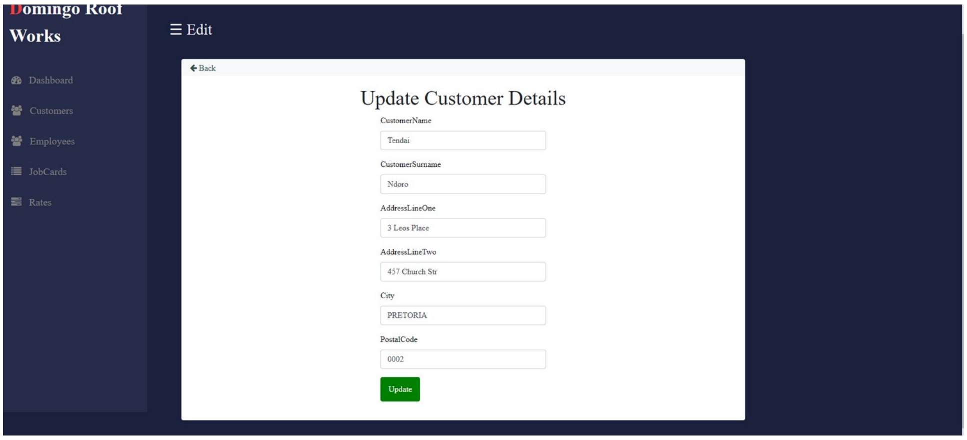
Job Card Number	Number of Days	Employees Assigned	Manage
10102	2	0	+Add Employee iView Invoice
10211	7	0	+Add Employee iView Invoice
10471	2	0	+Add Employee iView Invoice
13521	3	0	+Add Employee iView Invoice
13697	7	0	+Add Employee iView Invoice
14253	2	1	+Add Employee iView Invoice
12058	3	1	+Add Employee iView Invoice
10478	2	1	+Add Employee iView Invoice
11000	7	3	+Add Employee iView Invoice
11258	8	3	+Add Employee iView Invoice

The rate page allows one to update the price of a job type



Job Type	Rate	Manage
Floor Boarding	900	Edit
Semi Conversion	1080	Edit
Full Conversion	1200	Edit

ss



Update Customer Details

CustomerName: Tendai

CustomerSurname: Ndoro

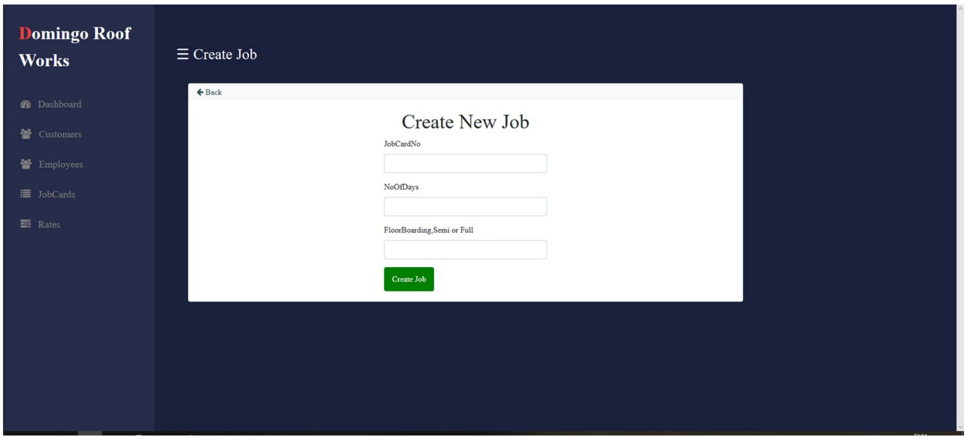
AddressLineOne: 3 Leos Place

AddressLineTwo: 457 Church Str

City: PRETORIA

PostalCode: 0002

The create new job page requires you to enter **FloorBoarding**, **Semi** or **Full** as the options in order to creae a job of those types

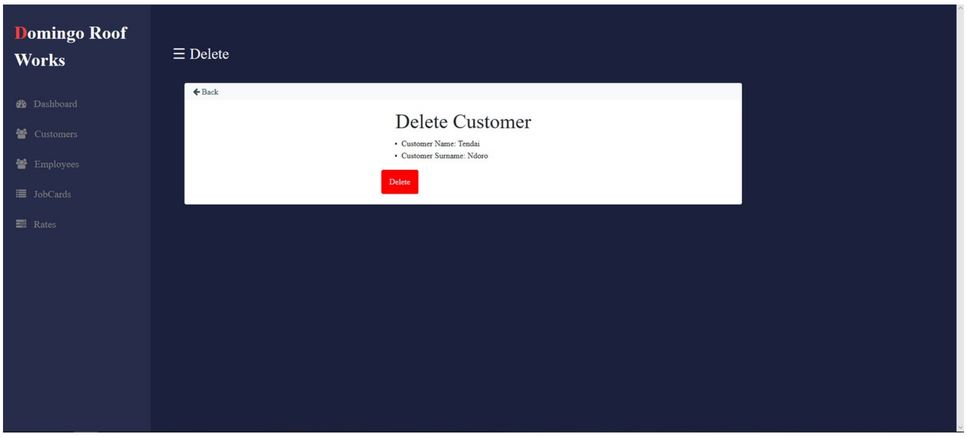


Create New Job

JobCardNo

NoOfDays

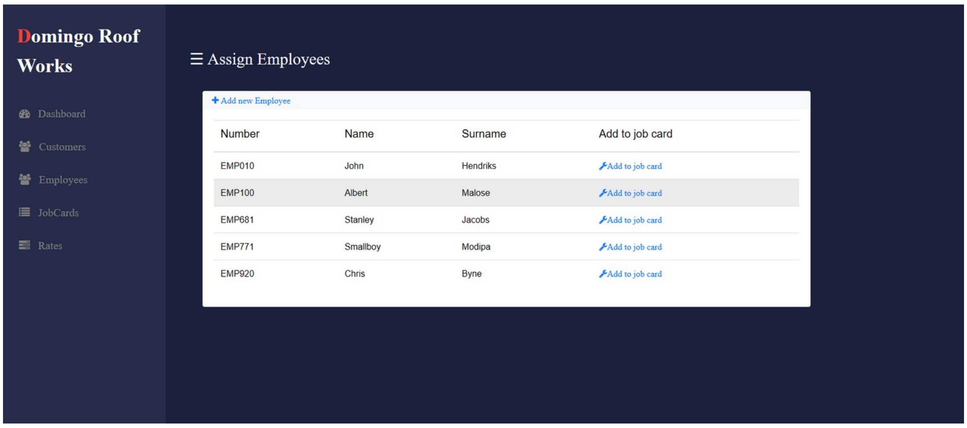
FloorBoarding,Semi or Full



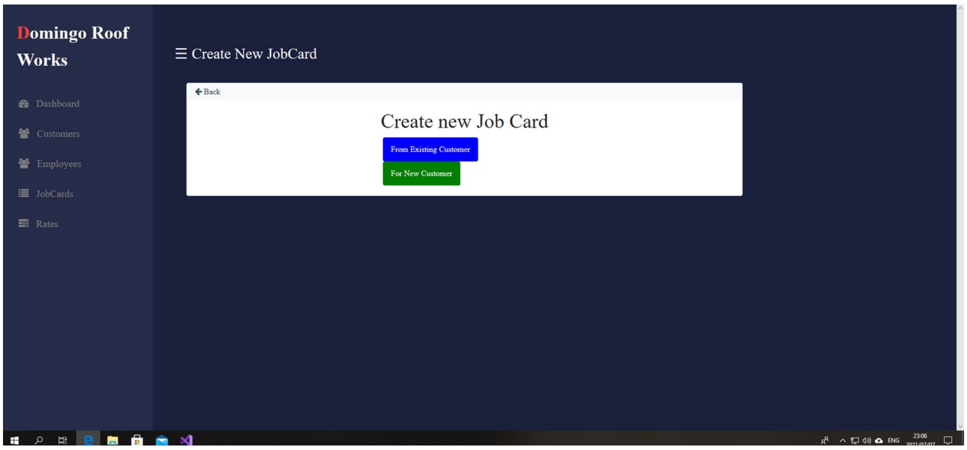
Delete Customer

- Customer Name: Tendai
- Customer Surname: Ndoro

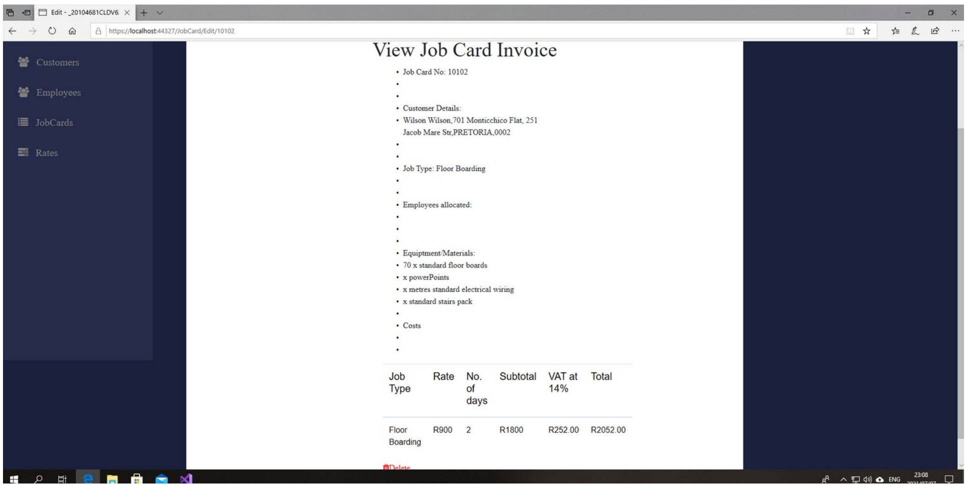
The assign employees page allows a an employee to be added to a job card, after the employee is added the user is directed to the Job Cards page where they can view an invoice of a job card.



Number	Name	Surname	Add to job card
EMP010	John	Hendriks	Add to job card
EMP100	Albert	Malose	Add to job card
EMP681	Stanley	Jacobs	Add to job card
EMP771	Smallboy	Modipa	Add to job card
EMP920	Chris	Byne	Add to job card



Create new Job Card



View Job Card Invoice

- Job Card No: 10102
- Customer Details:
- Wilson Wilson,701 Monticchico Flat, 251 Jacob Mare Str,PRETORIA,0002
- Job Type: Floor Boarding
- Employees allocated:
- Equiptment/Materials:
- 70 x standard floor boards
- x powerPoints
- x metres standard electrical wiring
- x standard stairs pack
- Costs

Job Type	Rate	No. of days	Subtotal	VAT at 14%	Total
Floor Boarding	R900	2	R1800	R252.00	R2052.00

4. Reflection

- Learning HTML and CSS is quite beneficial as now I feel as if I am capable of becoming a web developer.
- The most impactful learning unit would have to be where we covered advanced SQL concepts
- The videos provided under additional resources helped me create my website in particular
- Yes I will definitely use the skills i learnt in this course in my career as I can now create, read, update and delete database objects while also knowing how to connect databases to web apps
- I would not change anything from the past as I enjoy learning and pushing myself to be better.

5. Conclusion

As I have learnt, category discriminators exist to remove null values in a database and I decided to properly implement it into my task 2 which is why I currently have the Materials table as a category discriminator which comprises of the common materials in full conversion and semi conversion (Power points and Standard Electrical Wiring).

REFERENCE LIST

Anup, 2013. *SQL JOIN and different types of JOINs.* [Online]
Available at: https://stackoverflow.com/questions/17946221/sql-join-and-different-types-of-joins
[Accessed 6 April 2021].

Erkec, E., 2018. *How to use SQL Database Query Editor in Azure Portal.* [Online]
Available at: https://www.mssqltips.com/sqlservertip/5695/how-to-use-sql-database-query-editor-in-azure-portal/
[Accessed 11 May 2021].

Erkec, E., 2020. *How to UPDATE from a SELECT statement in SQL Server.* [Online]

Available at: https://www.sqlshack.com/how-to-update-from-a-select-statement-in-sql-server/
[Accessed 8 April 2021].

Mahajan, G., 2019. *Create Tables in SQL Server with T-SQL.* [Online]
Available at: https://www.mssqltips.com/sqlservertip/6197/create-tables-in-sql-server-with-tsql/
[Accessed 5 April 2021].

MBuschi, 2021. *azure db export to blob storage failed.* [Online]
Available at: https://dba.stackexchange.com/questions/281240/azure-db-export-to-blob-storage-failed
[Accessed 11 May 2021].

SQLSERVERTutorial.Net, 2019. *QL Server INSERT INTO SELECT.* [Online]
Available at: https://www.sqlservertutorial.net/sql-server-basics/sql-server-insert-into-select/
[Accessed 5 April 2021].

Stevestein, 2019. *Azure SQL Database and Managed Instance Import/Export service takes a long time to import or export a database.* [Online]
Available at: https://docs.microsoft.com/en-us/azure/azure-sql/database/database-import-export-hang
[Accessed 11 May 2021].

Taryn, 2012. *sql how to cast a select query.* [Online]
Available at: https://stackoverflow.com/questions/11989465/sql-how-to-cast-a-select-query
[Accessed 8 April 2021].

Tunggal, A. T., 2020. *Why is Cybersecurity Important?.* [Online]
Available at: https://www.upguard.com/blog/cybersecurity-important#:~:text=Cybersecurity%20is%20important%20because%20it,from%20theft%20and%20damage%20attempted
[Accessed 19 November 2020].

VanMSFT, 2020. *Azure SQL Database and Azure Synapse IP firewall rules.* [Online]

Available at: https://docs.microsoft.com/en-us/azure/azure-sql/database/firewall-configure
[Accessed 11 May 2021].

VisualParadigm, n.d. *Using Discriminator Column.* [Online]
Available at:
https://www.visual-paradigm.com/support/documents/vpuserguide/3563/3564/85389_usingdiscrim.html
[Accessed 3 May 2021].

w3resource, 2020. *SQL GROUP BY clause.* [Online]
Available at: https://www.w3resource.com/sql/group-by.php
[Accessed 9 April 2021].

w3resource, 2020. *SQL INNER JOIN.* [Online]
Available at: https://www.w3resource.com/sql/joins/perform-an-inner-join.php
[Accessed 5 April 2021].

WaLinke, 2017. *Count value in SQL [duplicate].* [Online]
Available at: https://stackoverflow.com/questions/44485870/count-value-in-sql?noredirect=1&lq=1
[Accessed 8 April 2021].