Managing & Programming Database MCDA5540

Master of Science in Computing and Data Analytics
Team Project Halifax Science Library (HSL)

Submitted by:
Caner Adil Irfanoglu (A00425840)
Sunil Padikar (A00428089)
Vinay Govindan (A00429120)
Gaganpreet Singh (A00429660)

Submitted to:
TRISHLA SHAH

Contents

EER Diagram:

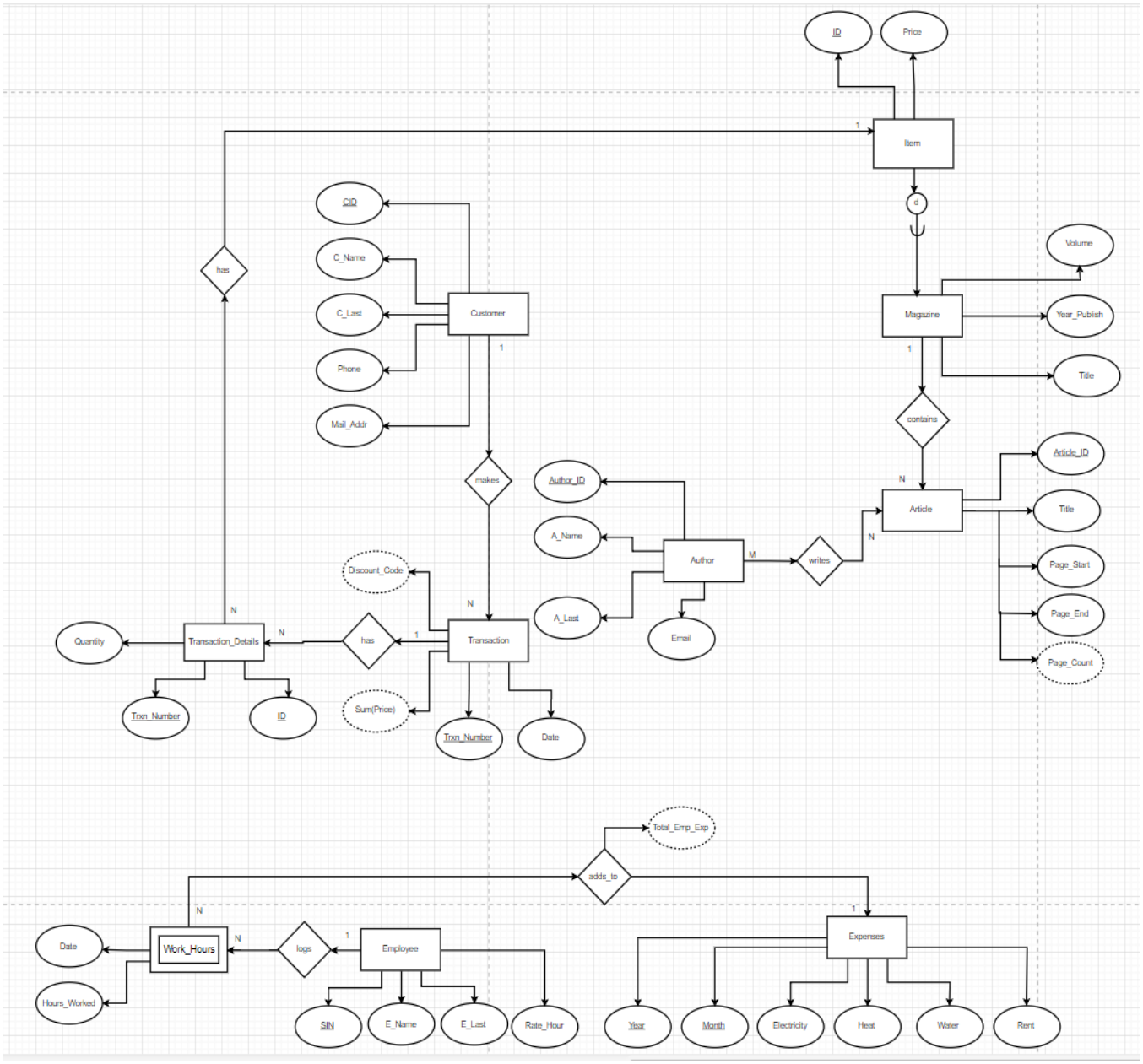


Figure 1 EER Diagram

Schema :

All the tables below are in 1NF as each filed has only one datatype, has only a single value per row and each row can be uniquely identified.

ITEM(ID, Price)
ITEM tables has no partial or transitive dependency as price is uniquely identified by the primary key ID and price cannot determined ID uniquely hence the relation is in BCNF.

Note: ID is unique for each vol. magazine and it is foreign key in MAGAZINE table as Magazine_ID

MAGAZINE(Maganize_ID unique NOT NULL , Title, Year_Publish, Volume)
Foreign Key MAGAZINE (Maganize_ID) references ITEM (ID)
Magazine is in 2NF and has no partial dependency as all the fields depends on both Title and Volume which put together forms the Candidate Key. It is in 3NF, BCNF as Maganize_ID and Year_Publish depends only on Title+ Volume & not the other way around hence there is no transitive dependency.

ARTICLE(Article_ID, Title, Page_Start, Page_End, Magazine_ID NOT NULL)
Foreign Key ARTICLE (Magazine_ID) references MAGAZINE (Magazine_ID)
The table Article is already in 3NF, BCNF as there is No partial and transitive dependencies since all the fields depend on Article_ID & not the other way around.

AUTHOR(Author_ID, A_Name, A_Last, Email)
The Author table is in 3NF, no partial and transitive dependencies as all the fields depend on Article_ID.

WRITES(Author_ID, Article_ID)
Foreign Key WRITES (Article_ID) references ARTICLE (Article_ID)
Foreign Key WRITES (Author_ID) references AUTHOR (Author_ID)
Writes table satisfies BCNF because both Author_ID and Article_ID form the candidate key and there is no partial or transitive dependency.

CUSTOMER(CID, C_Name, C_Last, Phone, Mail_Addr)
The Customer table is already in 3NF, BCNF as there is no partial and transitive dependencies since all the fields depend on CID and not the other way around.

TRANSACTION (Trxn_Number, Date, CID, Discount_Code)

Foreign Key TRANSACTION(CID) references CUSTOMER(CID)

The Transaction table is in 3NF, BCNF since there is no partial or transitive dependency and when a customer levels up their discount code will change even though CID remains the same. So, Discount Code is only determined by Trxn_Number and so are the other fields.

TRANSACTION_DETAILS (Trxn_Number, ID, Quantity)

Foreign Key TRANSACTION_DETAILS(Trxn_Number) references TRANSACTION(Trxn_Number)

Foreign Key TRANSACTION_DETAILS(ID) references ITEM(ID)

The Transaction_Details table is in 3NF, BCNF since there is no partial or transitive dependency and since a customer can purchase same items on different transactions Trxn_Number is required alongside with ID. Also, Trxn_Number is not sufficient to determine quantity, since the transaction can include same quantities of different items.
So, Quantity is only determined by using Trxn_Number and ID and not the other way around.

EMPLOYEE (SIN, E_Name, E_Last, Rate_Hour)

The Employee table is already is in 3NF, BCNF where there is no partial and transitive dependencies as all the fields depend on SIN and not the other way around.

WORK_HOURS (Date, SIN, Hours_Worked)

Foreign Key WORK_HOURS (SIN) references EMPLOYEE (SIN)

The WORK_HOURS table is already is in 3NF, BCNF where there is no partial and transitive dependencies as all the fields depend on Date and SIN and not the other way around.

EXPENSES (Year, Month, Electricity, Heat, Water, Rent, Total_Emp_Exp)

The Expenses table is already is in 3NF, BCNF as there is no partial and transitive dependencies since all the fields depend on Year and Month and not the other way around.

Current Status of tables :

Here we are just considering a sample of two tables “author” and “item” table to show its current state of query execution plan.

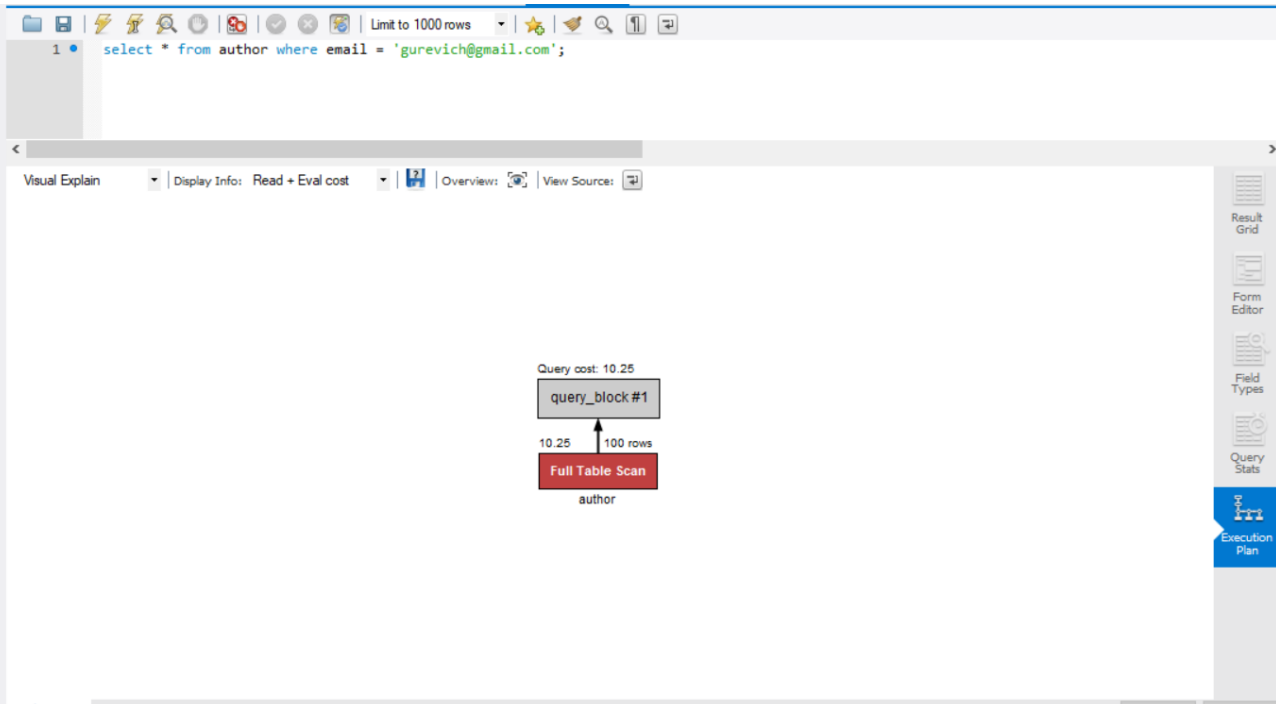


Figure 2 Query without optimization

The author table now upon querying for a author based on the email even thought it is logically unique to every author scans the entire table. Indexing the author table based on the email would reduce the number of scans.

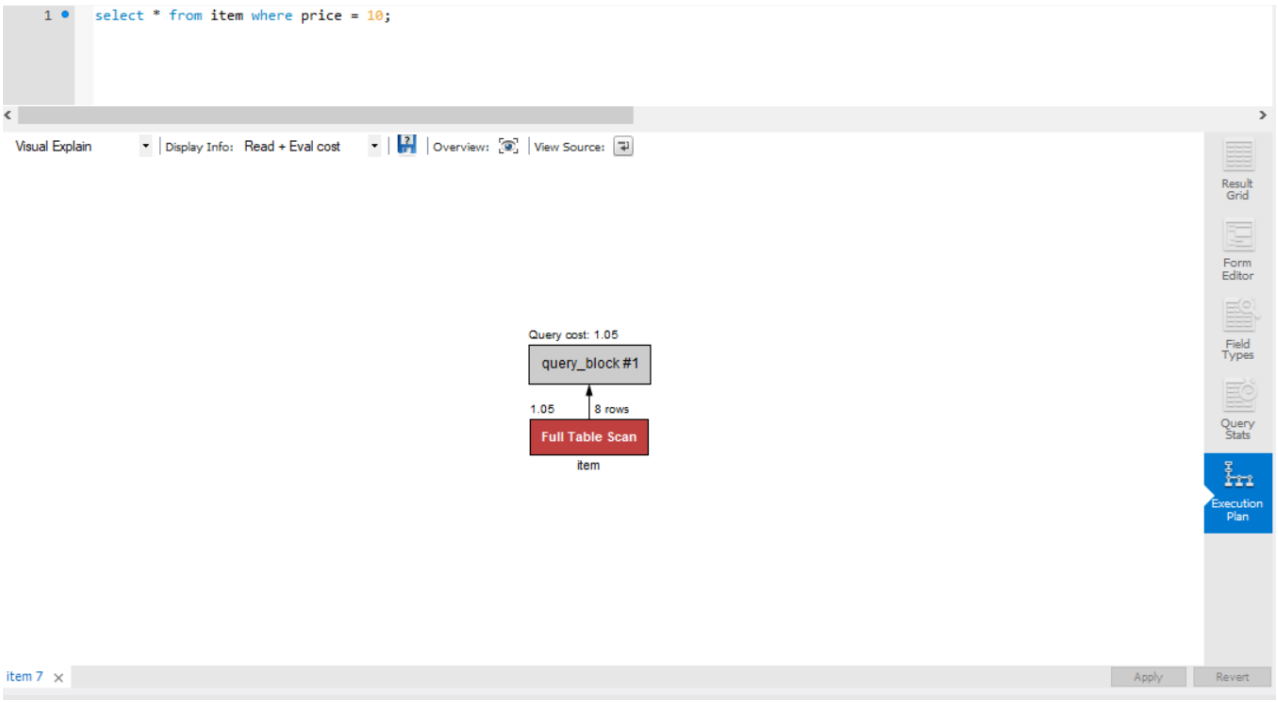


Figure 3 Query after Optimization

Similarly the item table can also be indexed based on the price which would reduce the number of rows being scanned.

Data Load Scripting workflow

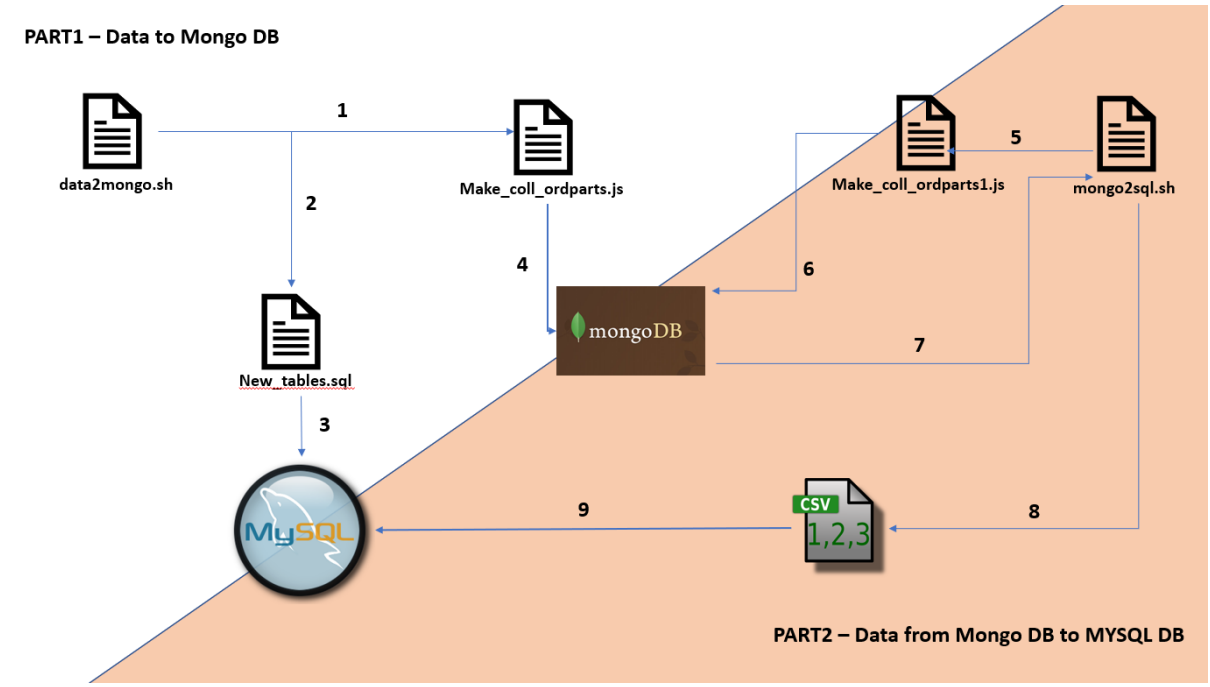


Figure 4 Data Flow Diagram

Data2mongo.sh:

Executes new_tables.sql which contains new schema of our database and loads data from article.json file to ARTICLE_PRE collection. It also executes Make_coll_ordparts.js to create WRITES and AUTHOR collections in the mongoDB.

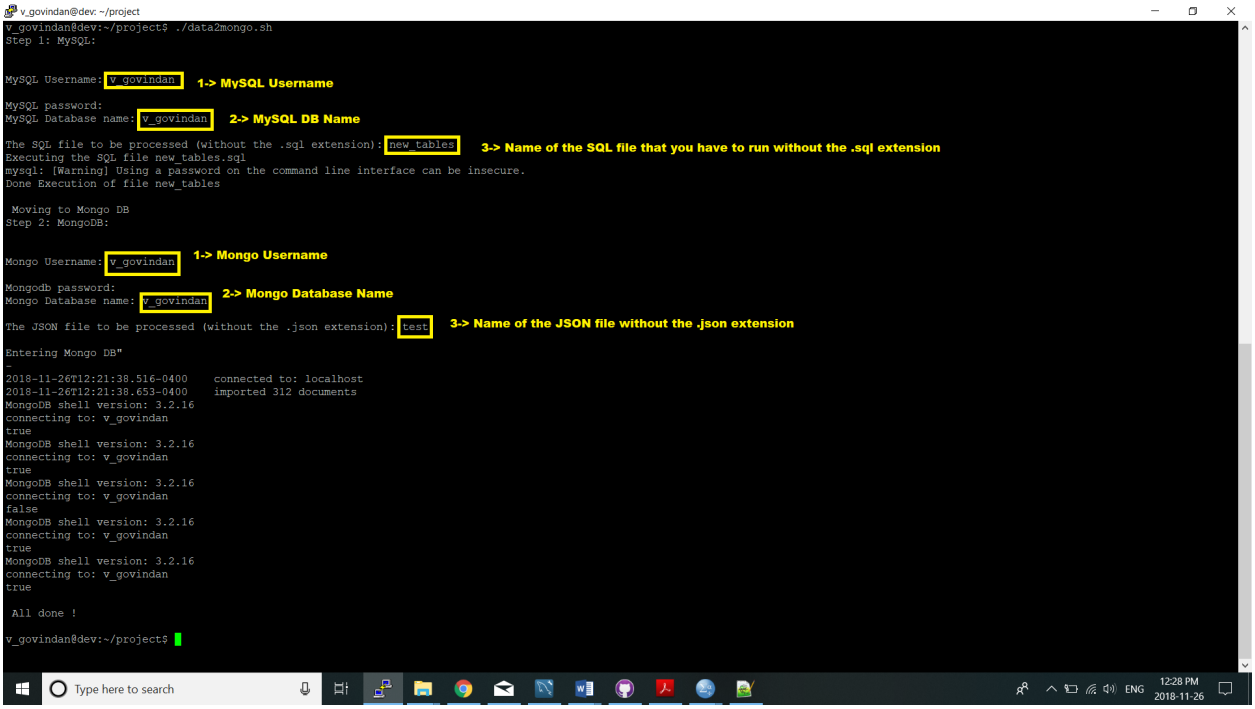


Figure 5 Data2mongo script working

Follow the below steps while running:

1. In the location ./data2mongo.sh to run the script
2. Enter your mysql credentials when prompted.
3. Enter the name of the database.
4. Specify the sql file to be run without .sql extension in this case “new_tables”
5. Enter your mongo credentials when prompted.
6. Enter the name of the mongo database required.
7. Specify the json file to be run without .json extension in this case “test”(which is a snippet of the articles.json).

Make_coll_ordparts.js:

Reads data from ARTICLE_PRE collection and parses it and loads into AUTHOR, WRITES collections.

mongo2sql.sh:

Executes Make_coll_orderparts1.js to create collections according to our schema (ARTICLE, AUTHOR, ITEM, WRITES) in the mongoDB. It also exports the data in new collection to respective csv files. Then, imported data from csv to sql tables and Creates indexes according to search criteria in our php application.

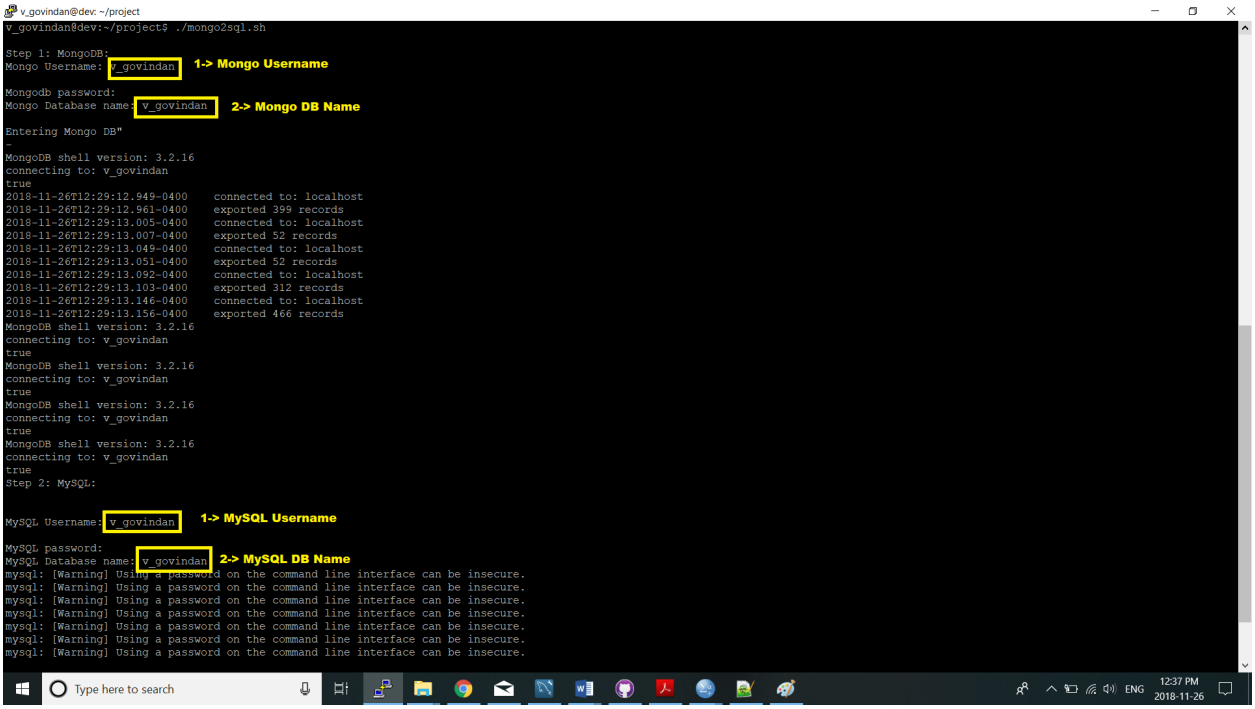


Figure 6 mongo2sql script workflow

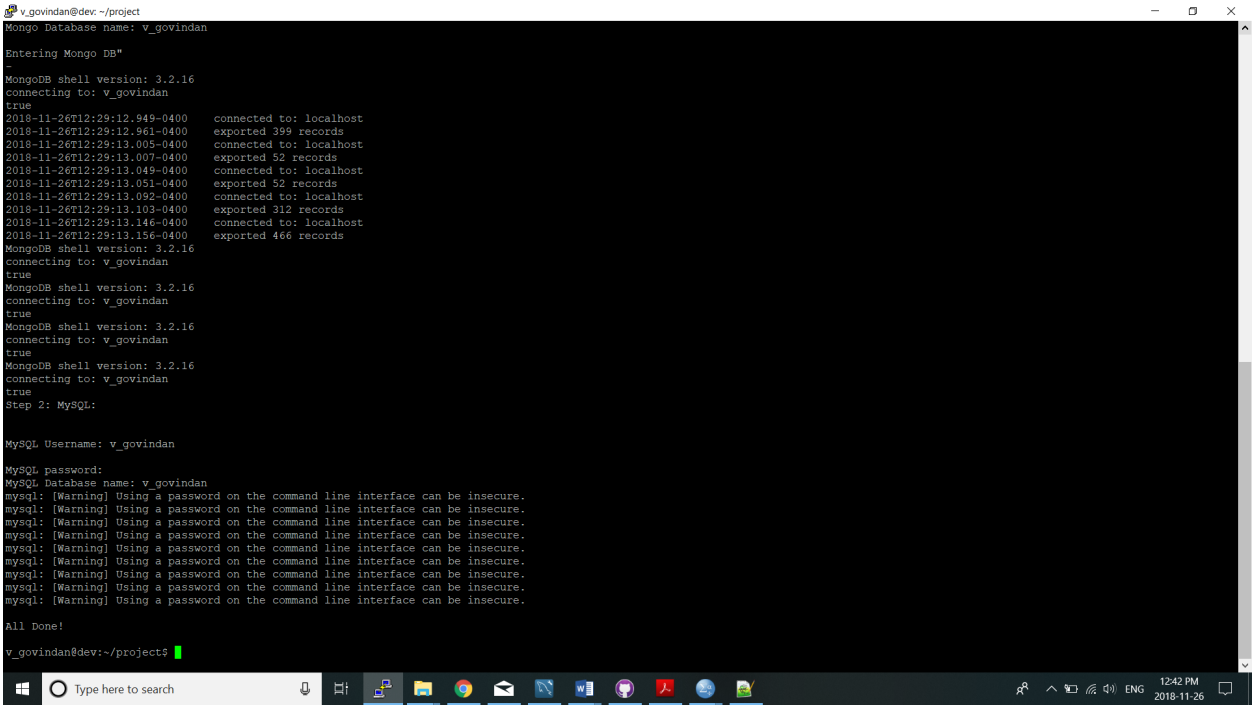


Figure 7 mongo2sql script workflow Continue

Follow the below steps while running:

1. In the location ./mongo2sql.sh to run the script
2. Enter your mongo credentials when prompted.
3. Enter the name of the mongo database required.
4. Enter your mysql credentials when prompted.
5. Enter the name of the database.

Make_coll_orderparts1.js:

Reads data from ARTICLE collection and creates ARTI, MAGAZINE, ITEM according to our schema.

Php Application workflow

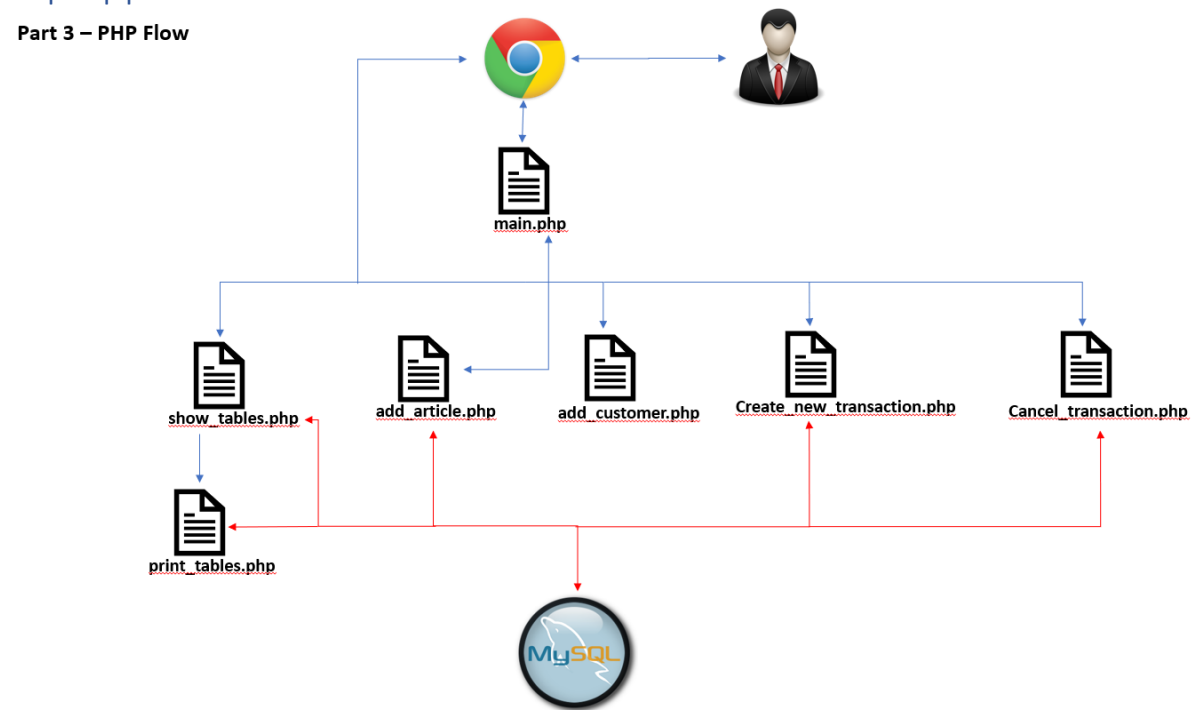


Figure 8 Php Application workflow

main.php:

Main page can be accessed from the below link:

http://dev.cs.smu.ca/~v_govindan/proj/main.php

It's our home page and has navigation links to all other pages like Show tables, Add Article, Add Customer, Add Transaction, Cancel Transaction.

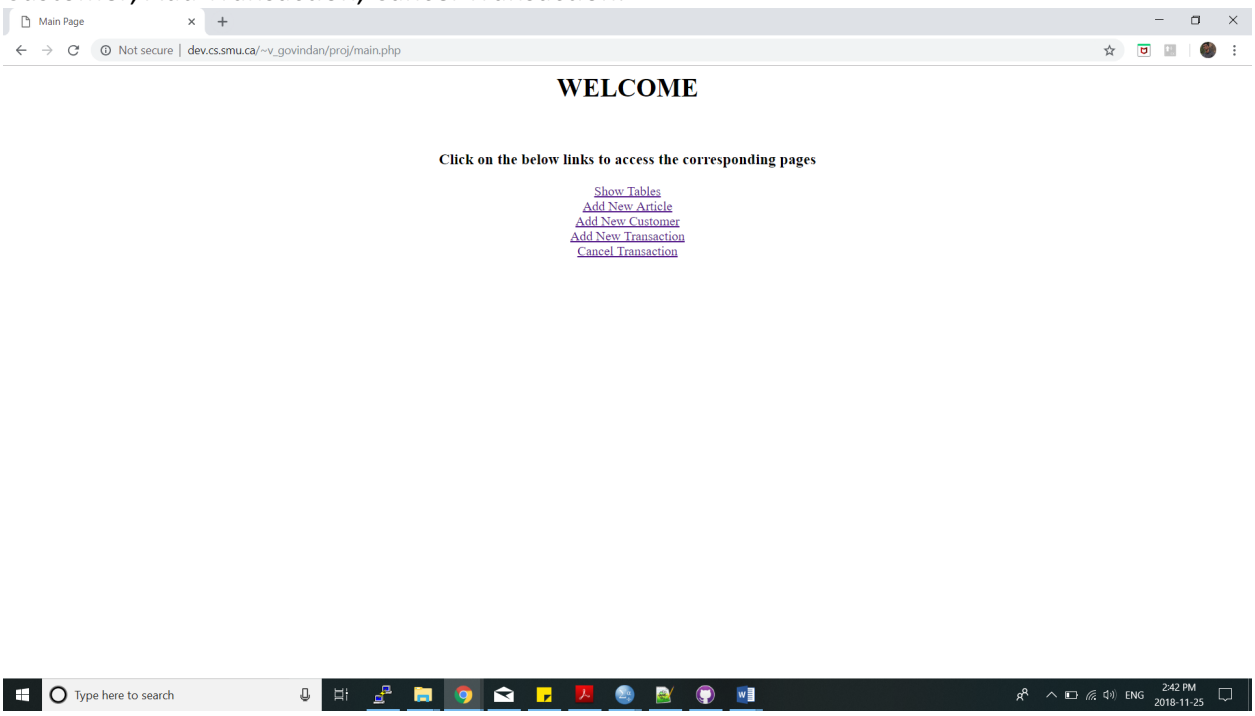


Figure 9 Main page

add_article.php:

Article Title, Page Start, Page End, Magazine, Authors are the inputs from the users. On submit it successfully adds new article to the database.

Figure 10 Add Article page

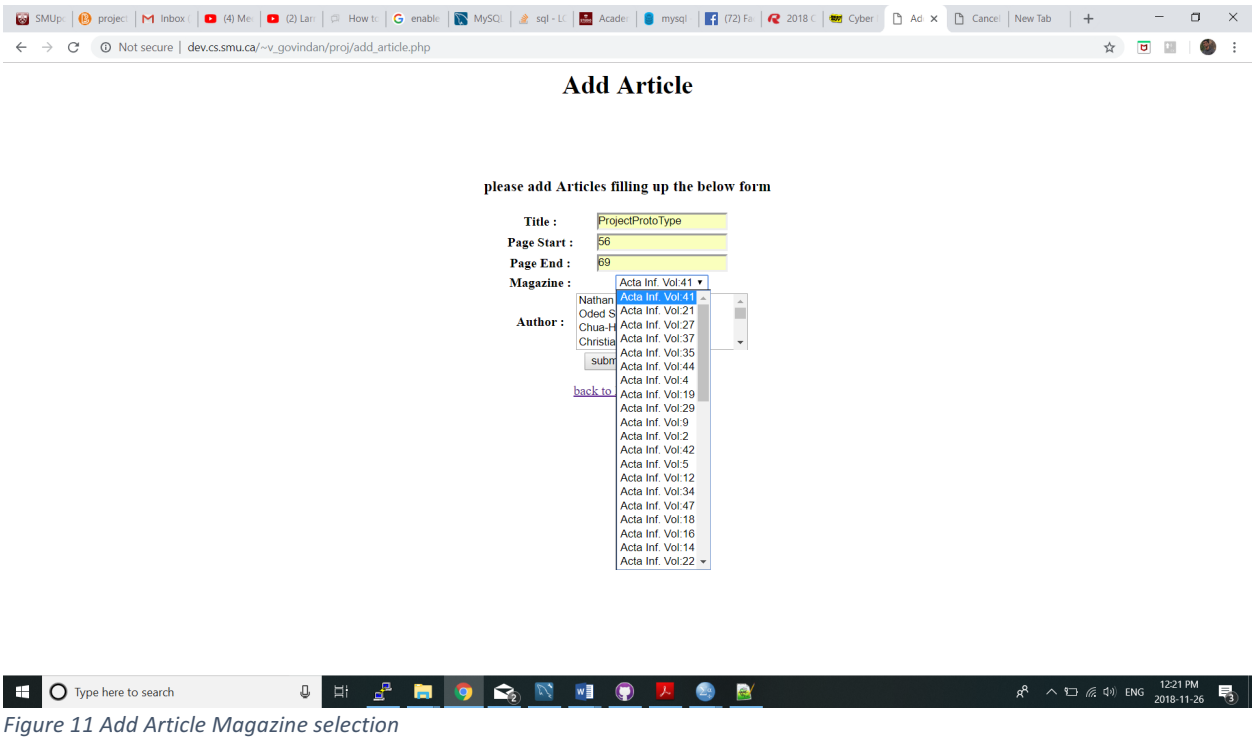


Figure 11 Add Article Magazine selection

Figure 12 Add Article Author selection

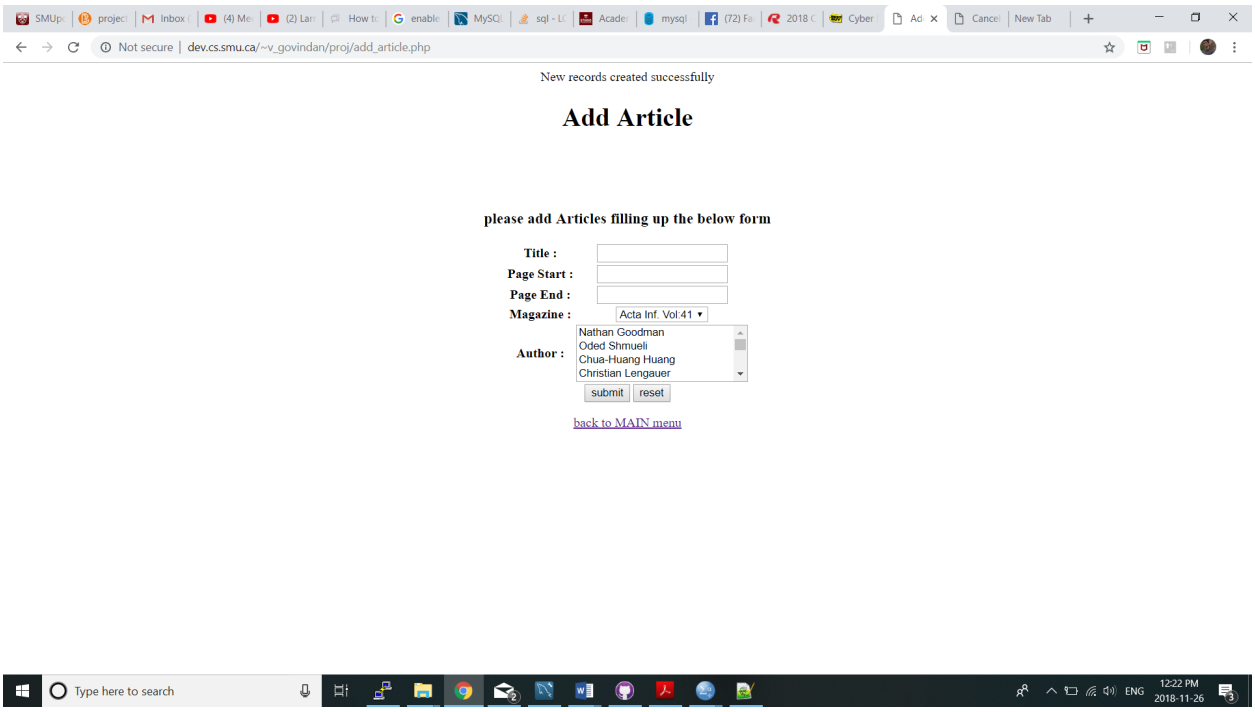


Figure 13 Add Article Successful

add_customer.php:

First Name, Last Name, Mail Address, Mobile Number are the inputs from the users. On submit it successfully adds new Customer to the database.

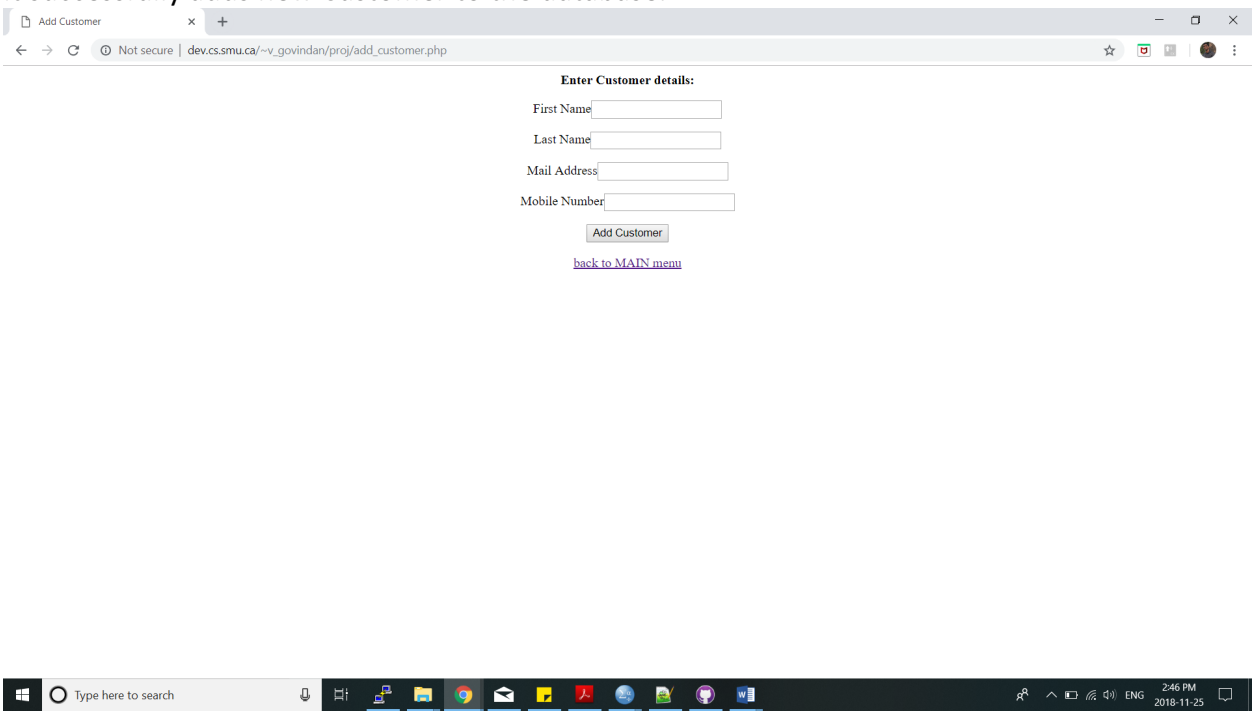


Figure 14 Add new Customer

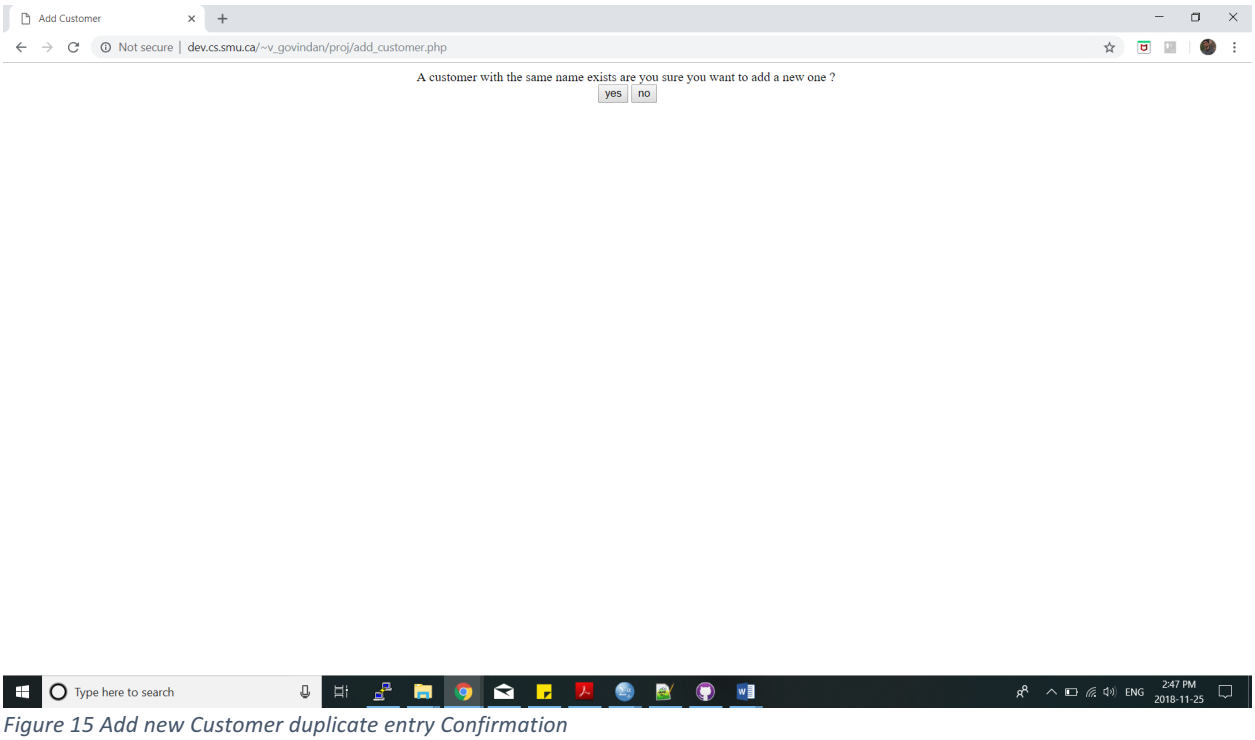


Figure 15 Add new Customer duplicate entry Confirmation

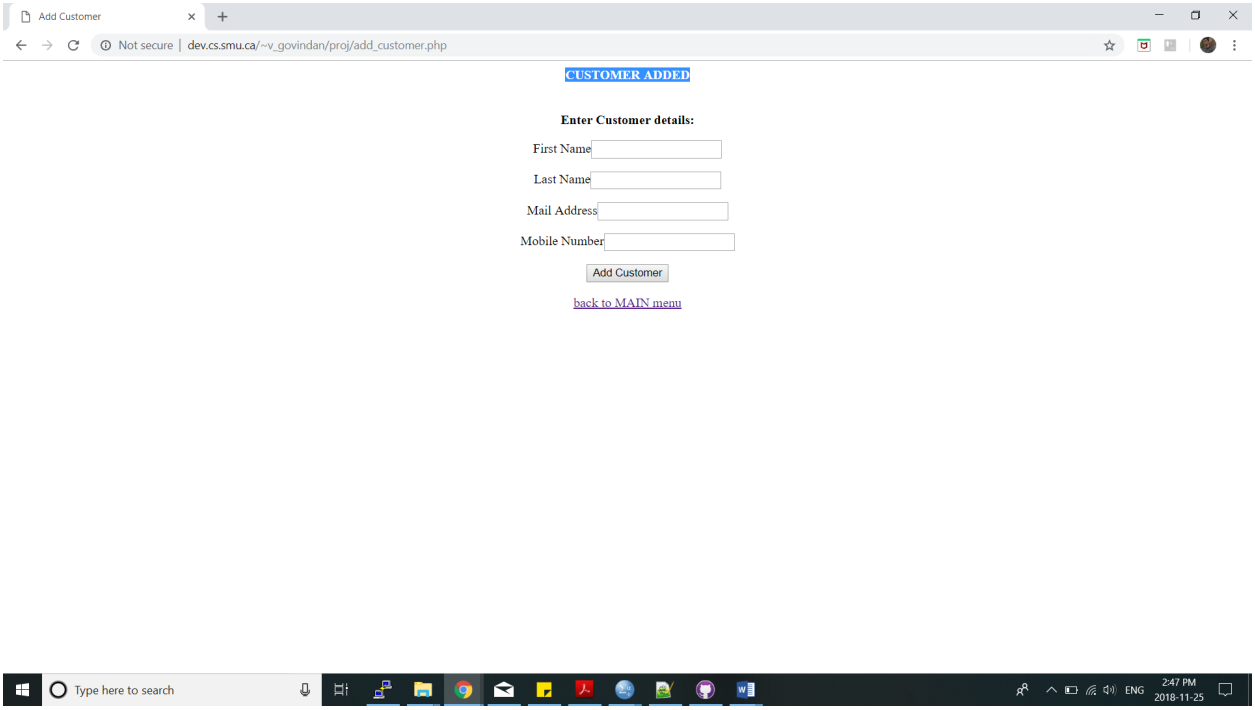


Figure 16 Add new Customer Successful

Create_new_transaction.php:

Customer Number, MagazineID are the inputs from the users. On submit it successfully adds new Transaction and Transaction details to the database. This page also calculates Discount code dynamically while creating new transaction.

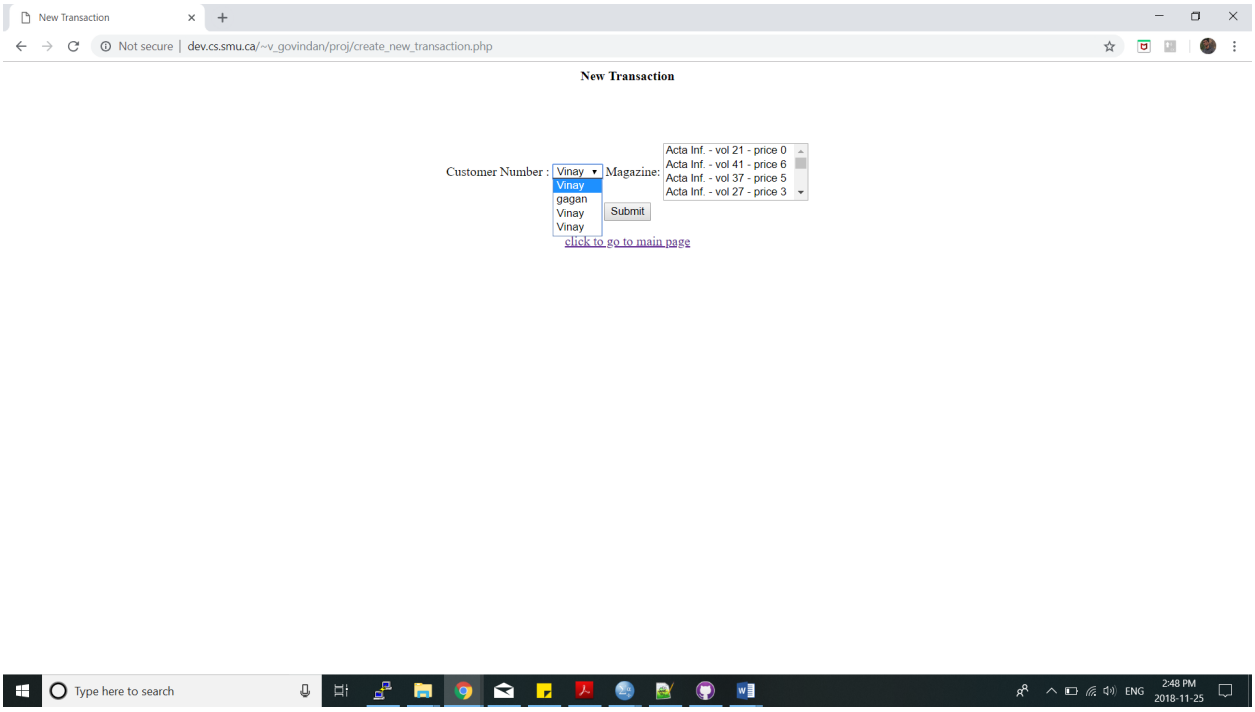


Figure 17 Add new Transaction

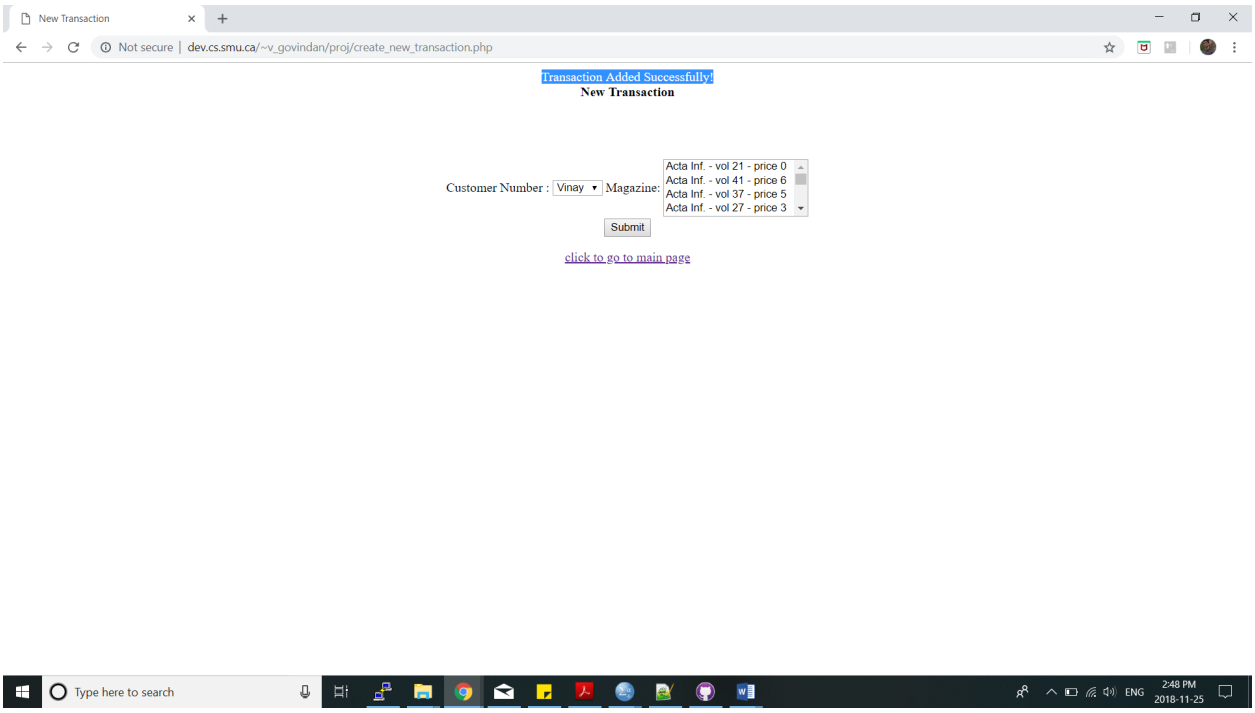


Figure 18 Add new Transaction Successful

Cancel_transaction.php:

Transaction Number is the input from the users. On submit it removes entries for given Transaction Number from Transaction and Transaction Details tables. If given transaction is older than 30 days, transaction cannot be cancelled.

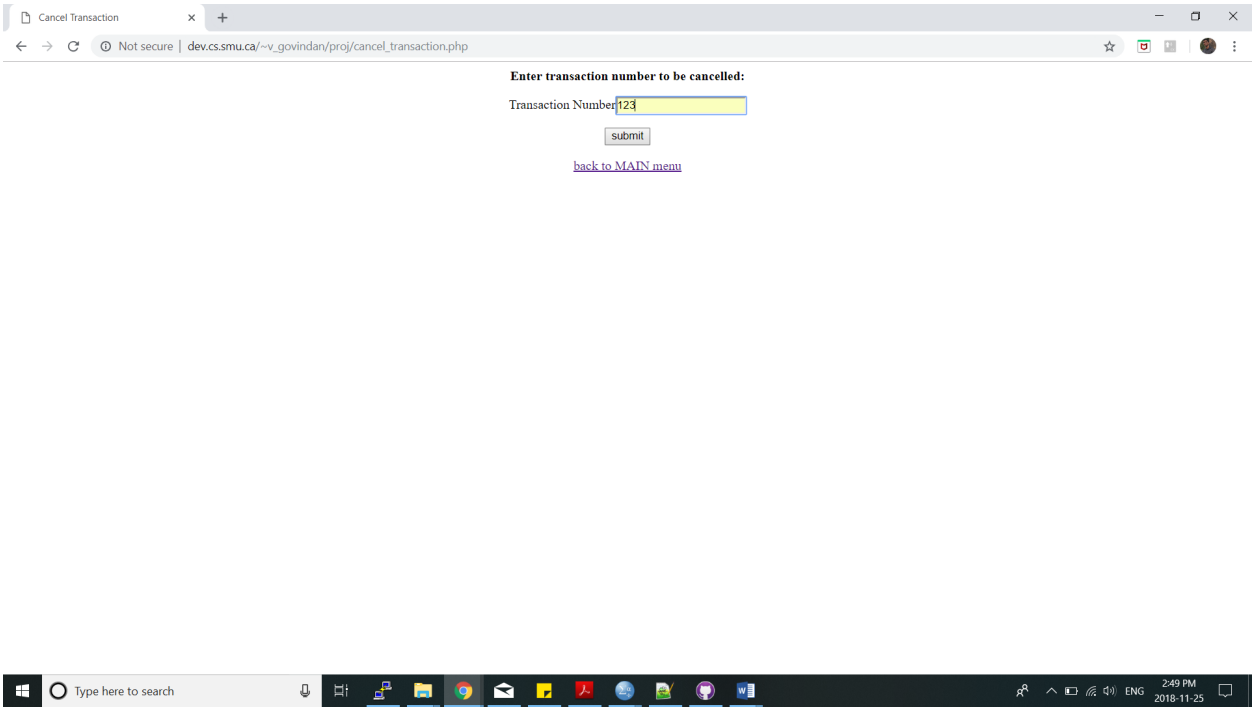


Figure 19 Cancel Transaction

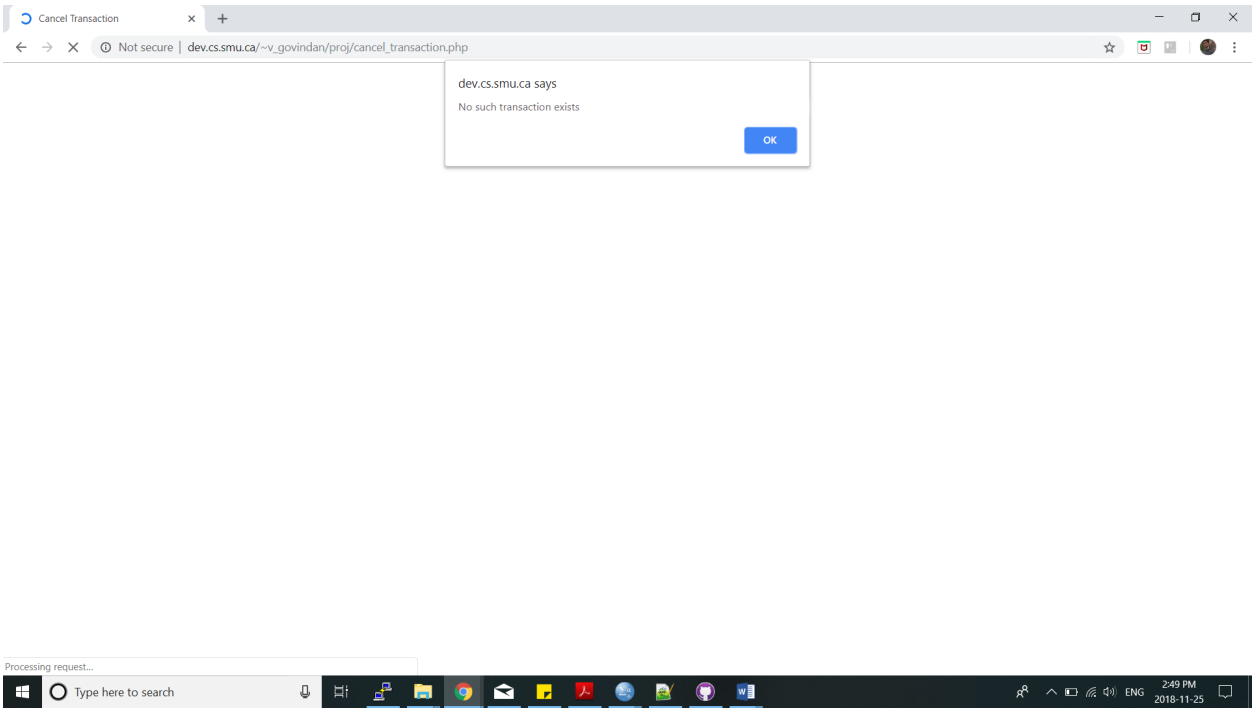


Figure 20 Cancel Transaction Invalid ID Warning

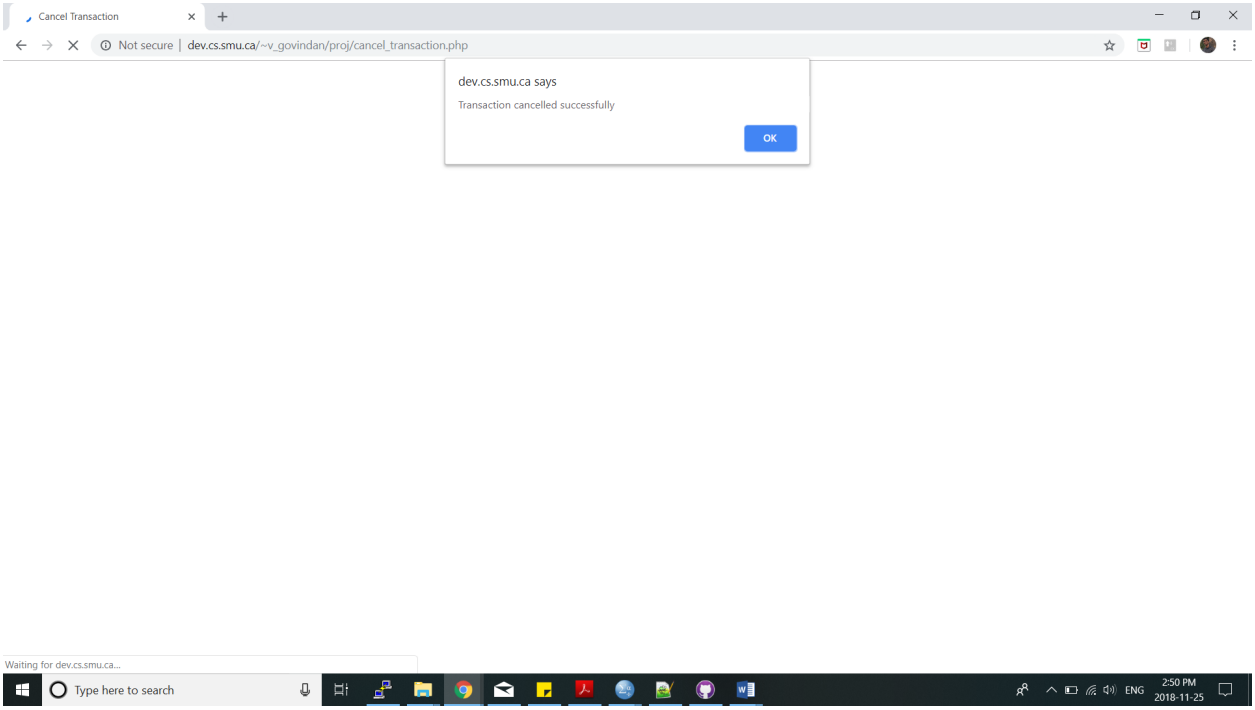


Figure 21 Cancel Transaction Successful

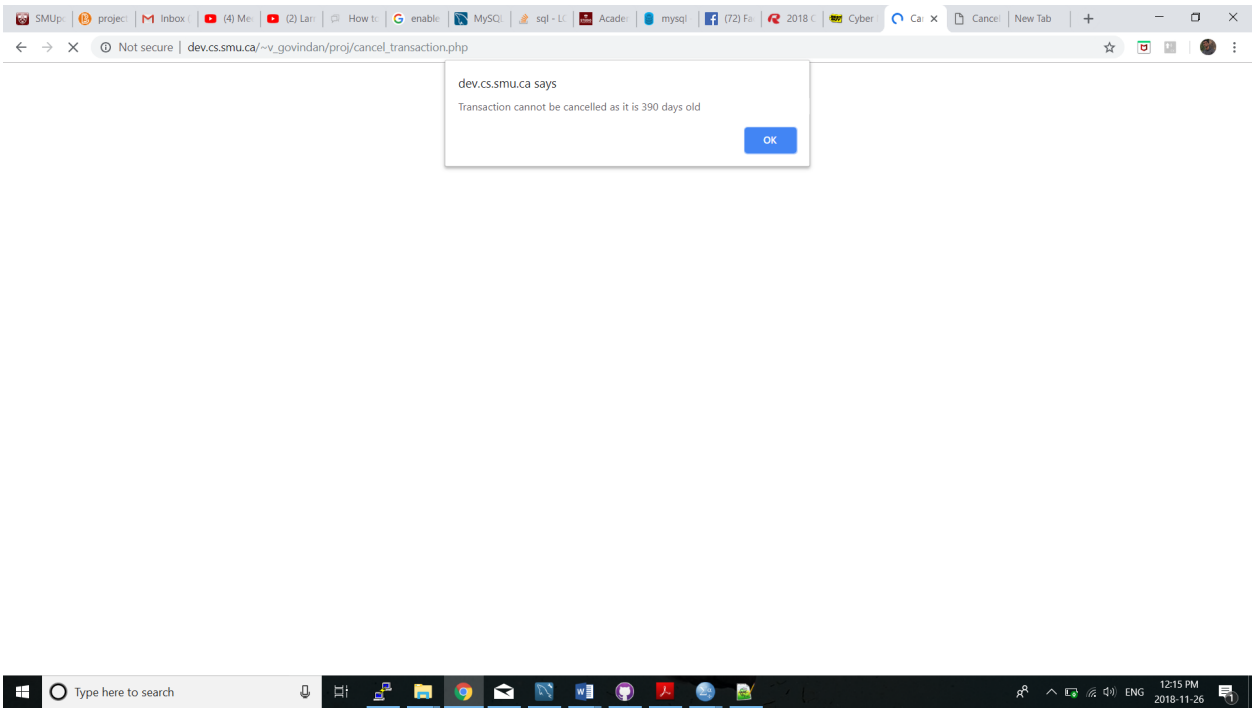


Figure 22 Cancel Transaction Error if transaction is older than 30 days

show_tables.php:

Displays all the table names in a database. Search for table names. For the Transaction table total amount displayed is not a field of Transaction table but, is result of join query.

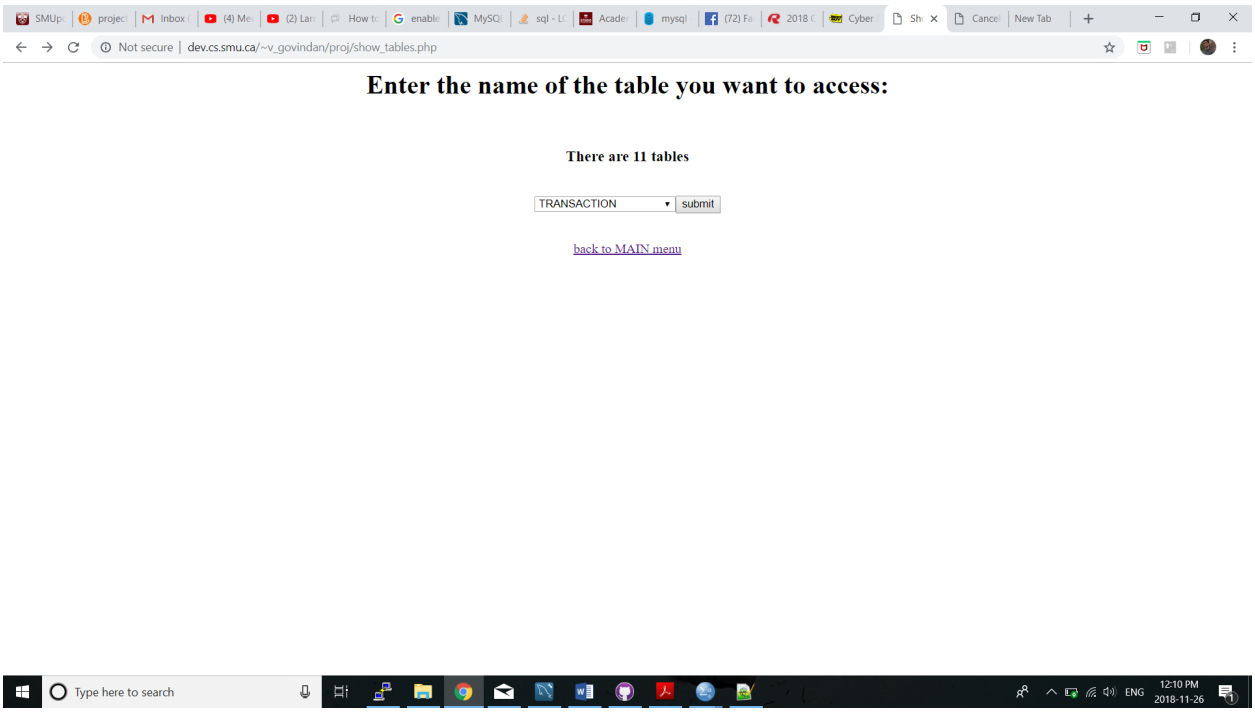
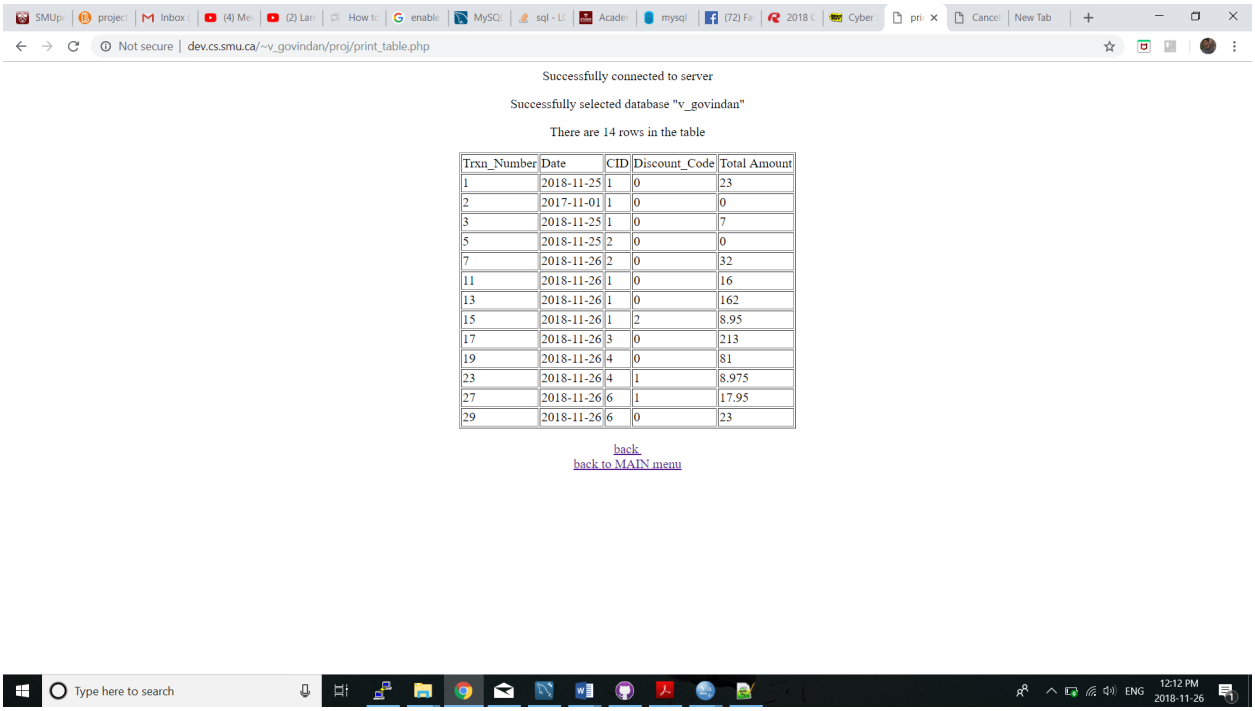


Figure 23 Show Tables

print_tables.php:

Displays data of the selected tables.



Successfully connected to server

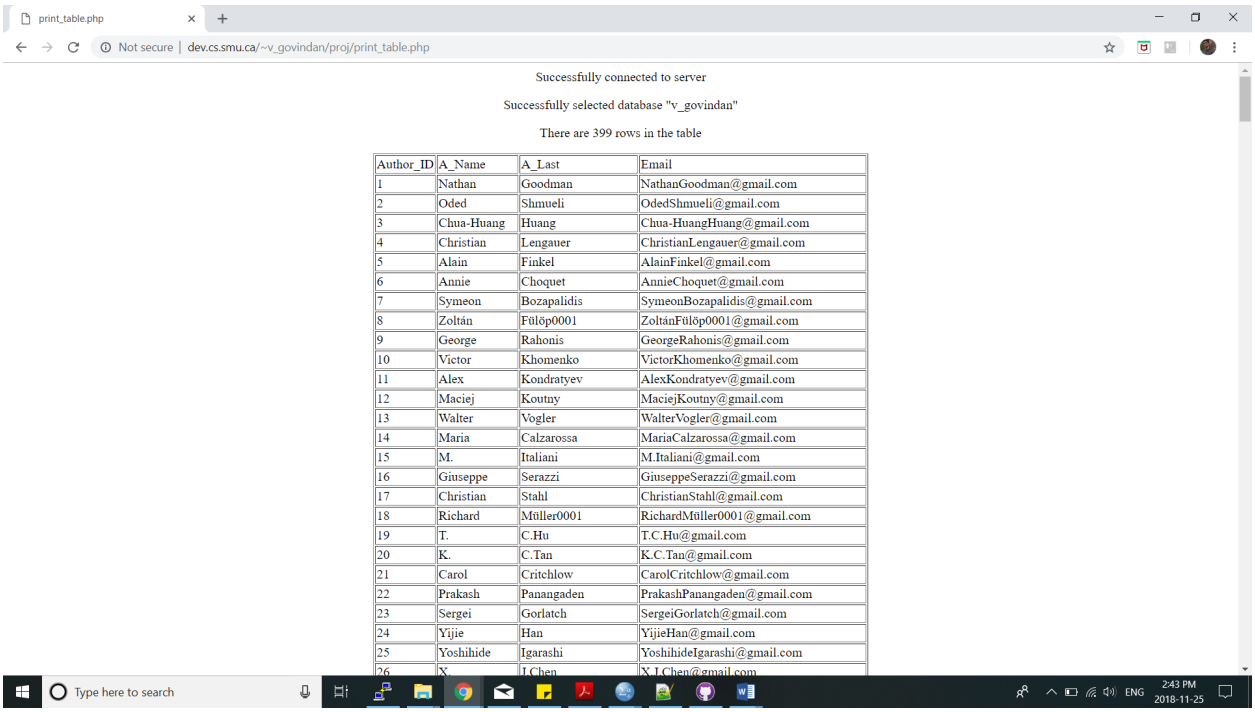
Successfully selected database "v_govindan"

There are 14 rows in the table

Trxn_Number	Date	CID	Discount_Code	Total Amount
1	2018-11-25	1	0	23
2	2017-11-01	1	0	0
3	2018-11-25	1	0	7
5	2018-11-25	2	0	0
7	2018-11-26	2	0	32
11	2018-11-26	1	0	16
13	2018-11-26	1	0	162
15	2018-11-26	1	2	8.95
17	2018-11-26	3	0	213
19	2018-11-26	4	0	81
23	2018-11-26	4	1	8.975
27	2018-11-26	6	1	17.95
29	2018-11-26	6	0	23

back
back to MAIN menu

Figure 24 Print Transaction table



Successfully connected to server

Successfully selected database "v_govindan"

There are 399 rows in the table

Author_ID	A_Name	A_Last	Email
1	Nathan	Goodman	NathanGoodman@gmail.com
2	Oded	Shmueli	OdedShmueli@gmail.com
3	Chua-Huang	Huang	Chua-HuangHuang@gmail.com
4	Christian	Lengauer	ChristianLengauer@gmail.com
5	Alain	Finkel	AlainFinkel@gmail.com
6	Annie	Choquet	AnnieChoquet@gmail.com
7	Symeon	Bozapalidis	SymeonBozapalidis@gmail.com
8	Zoltán	Fülöp0001	ZoltánFülöp0001@gmail.com
9	George	Rahonis	GeorgeRahonis@gmail.com
10	Victor	Khomenko	VictorKhomenko@gmail.com
11	Alex	Kondratyev	AlexKondratyev@gmail.com
12	Maciej	Koutny	MaciejKoutny@gmail.com
13	Walter	Vogler	WalterVogler@gmail.com
14	Maria	Calzarossa	MariaCalzarossa@gmail.com
15	M.	Italiani	M.Italiani@gmail.com
16	Giuseppe	Serazzi	GiuseppeSerazzi@gmail.com
17	Christian	Stahl	ChristianStahl@gmail.com
18	Richard	Müller0001	RichardMüller0001@gmail.com
19	T.	C.Hu	T.C.Hu@gmail.com
20	K.	C.Tan	K.C.Tan@gmail.com
21	Carol	Critchlow	CarolCritchlow@gmail.com
22	Prakash	Panangaden	PrakashPanangaden@gmail.com
23	Sergei	Gorlatch	SergeiGorlatch@gmail.com
24	Yijie	Han	YijieHan@gmail.com
25	Yoshihide	Igarashi	YoshihideIgarashi@gmail.com
26	X.	J.Chen	X.J.Chen@gmail.com

Figure 25 Print Author Table

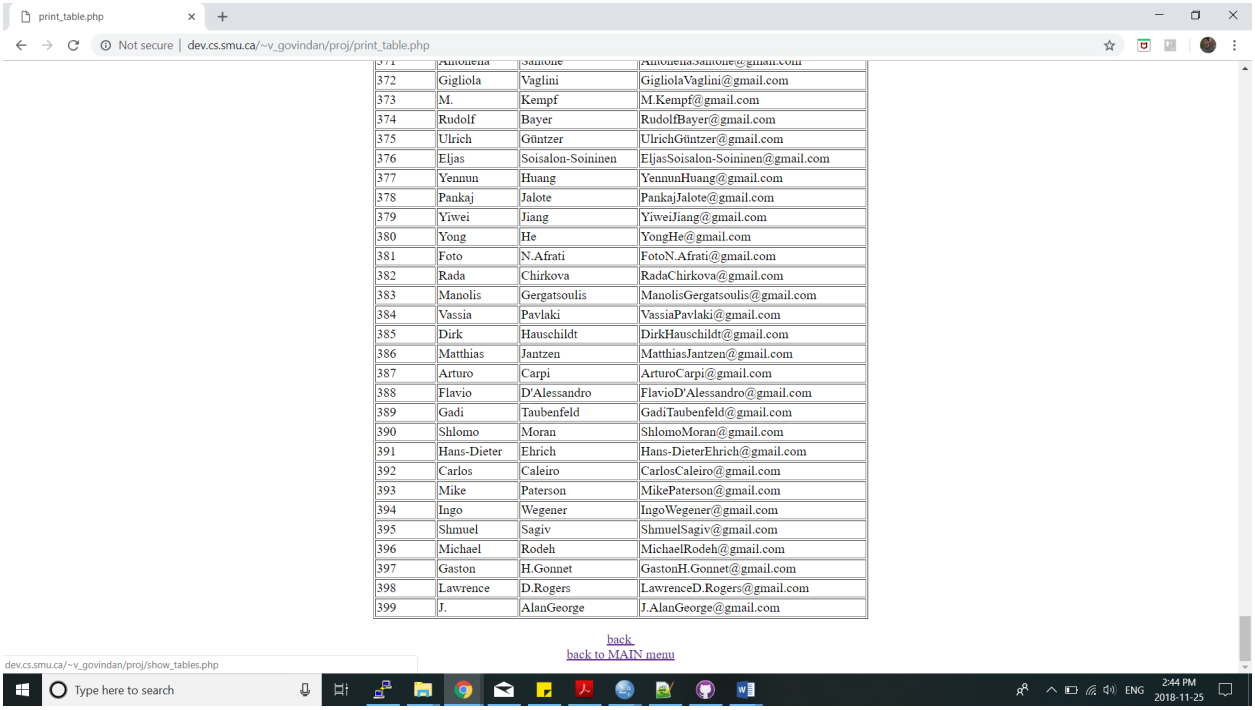


Figure 26 Print Author Table Continue

References:

1. w3schools
2. stackoverflow
3. For PHP files tutorial 6 material.
4. Computing and Data Analytics 5540 -- Team Project.pdf
5. JavaScript, Shell script have referenced tutorial 8 material