

**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: 3](#_Toc531107374)

[Schema : 4](#_Toc531107375)

[Current Status of tables : 6](#_Toc531107376)

[Data Load Scripting workflow 8](#_Toc531107377)

[Data2mongo.sh: 8](#_Toc531107378)

[Make\_coll\_ordparts.js: 9](#_Toc531107379)

[mongo2sql.sh: 9](#_Toc531107380)

[Make\_coll\_orderparts1.js: 10](#_Toc531107381)

[Php Application workflow 11](#_Toc531107382)

[main.php: 12](#_Toc531107383)

[add\_article.php: 13](#_Toc531107384)

[add\_customer.php: 15](#_Toc531107385)

[Create\_new\_transaction.php: 17](#_Toc531107386)

[Cancel\_transaction.php: 19](#_Toc531107387)

[show\_tables.php: 22](#_Toc531107388)

[print\_tables.php: 23](#_Toc531107389)

[References: 25](#_Toc531107390)

# EER Diagram:

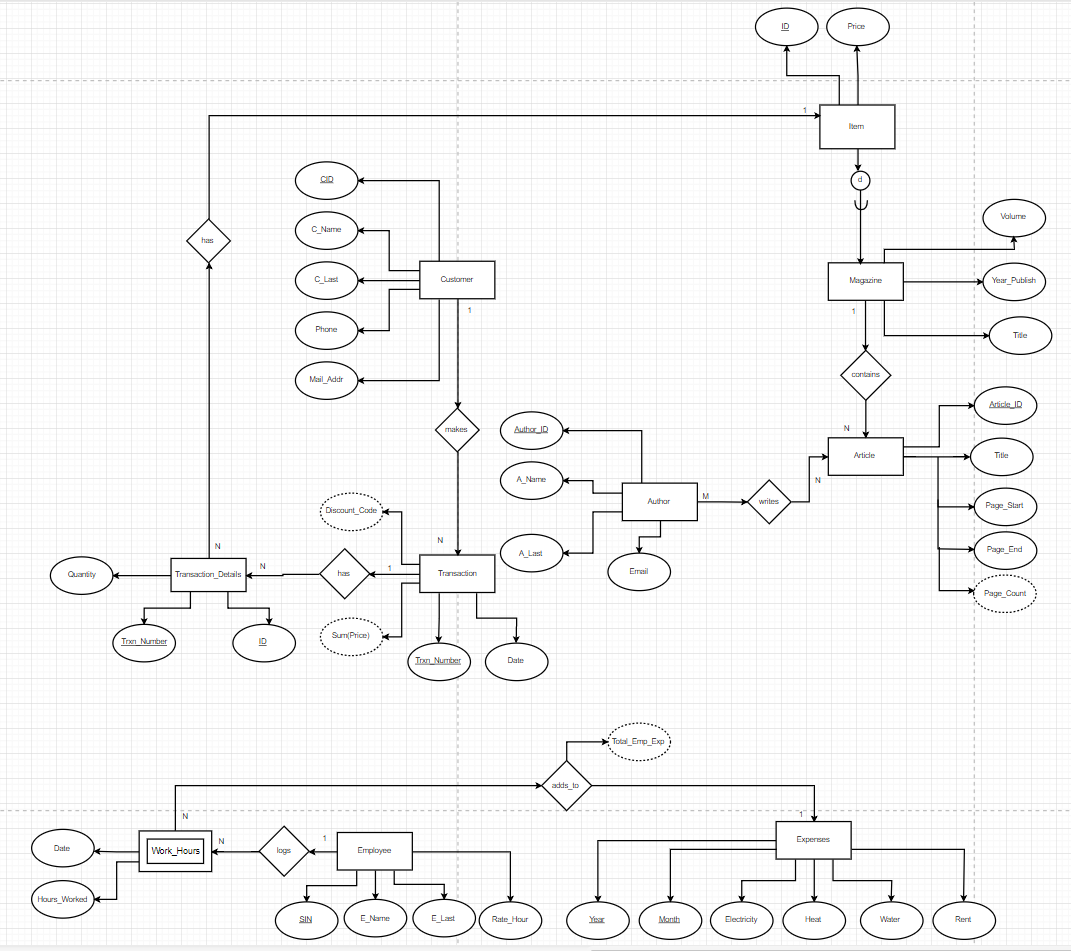


Figure EER Diagram

# 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.

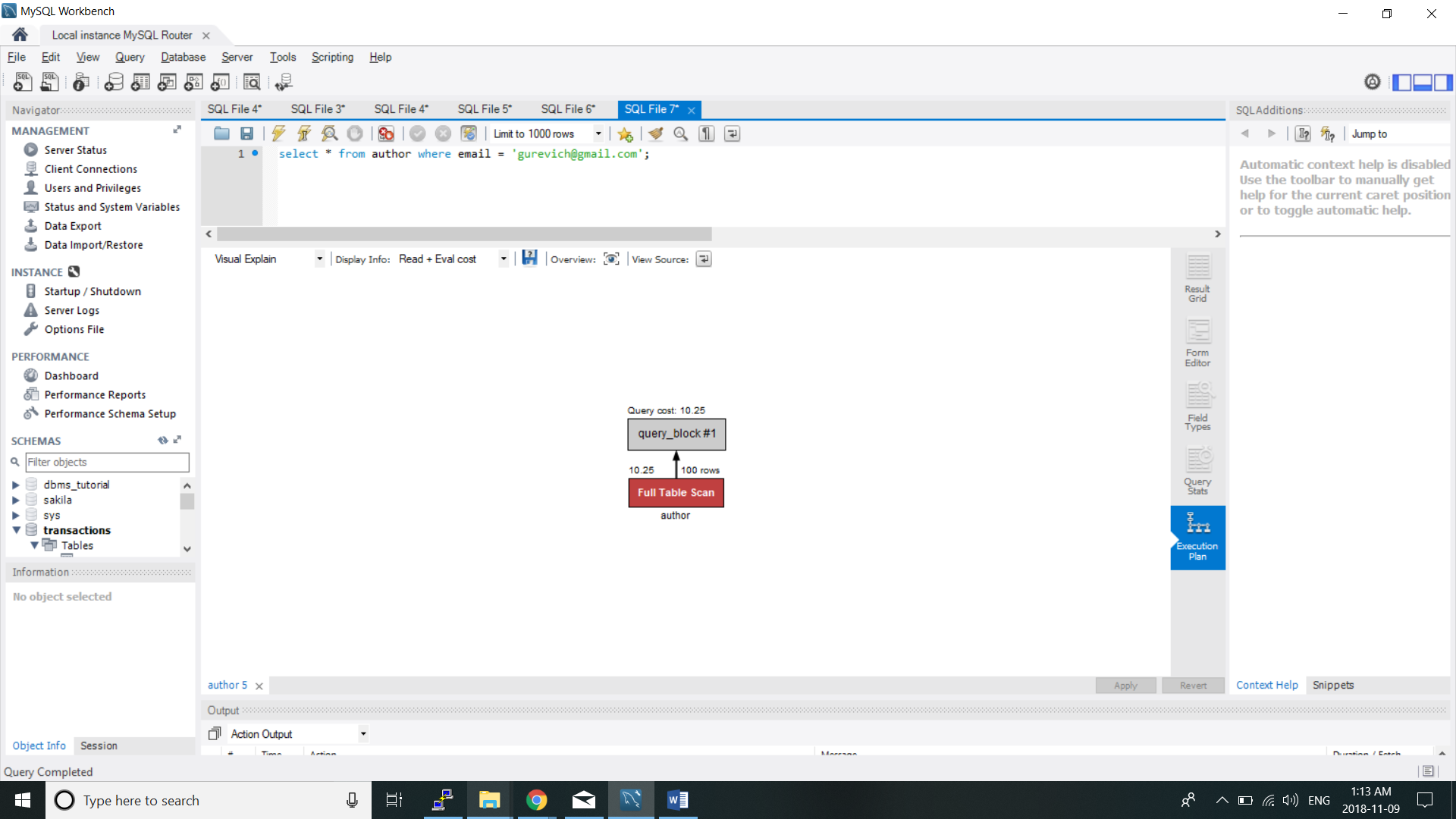


Figure Query without optimization

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.

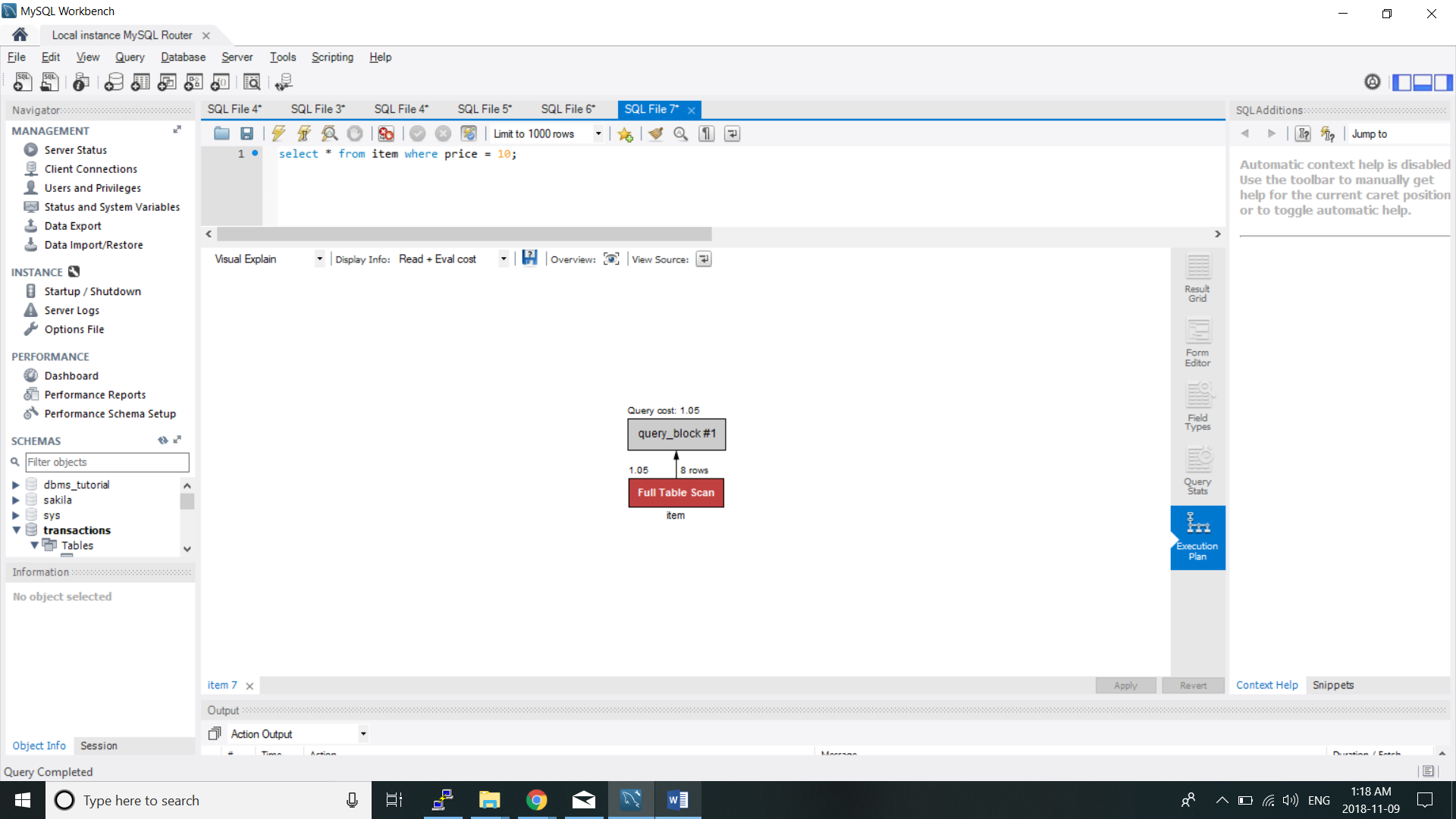


Figure Query after Optimization

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

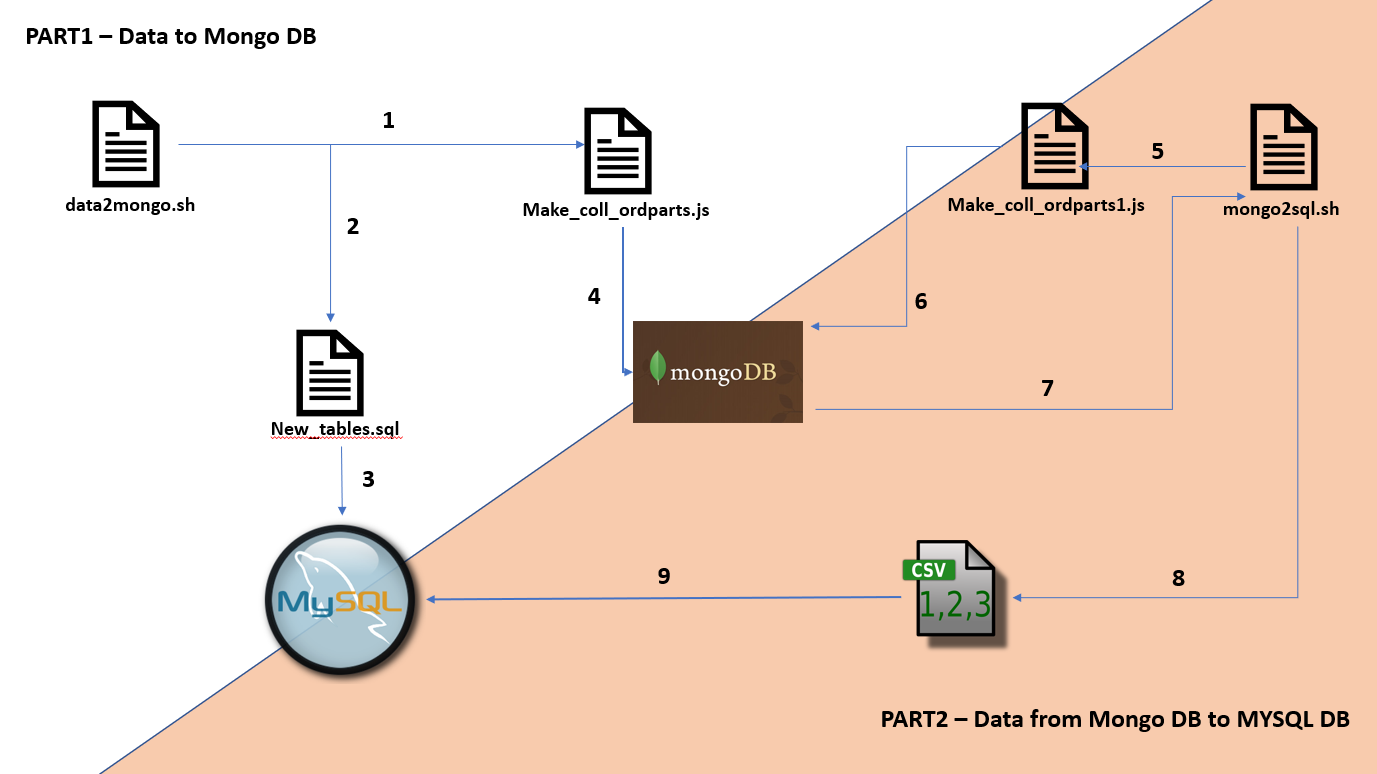


Figure Data Flow Diagram

## 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.

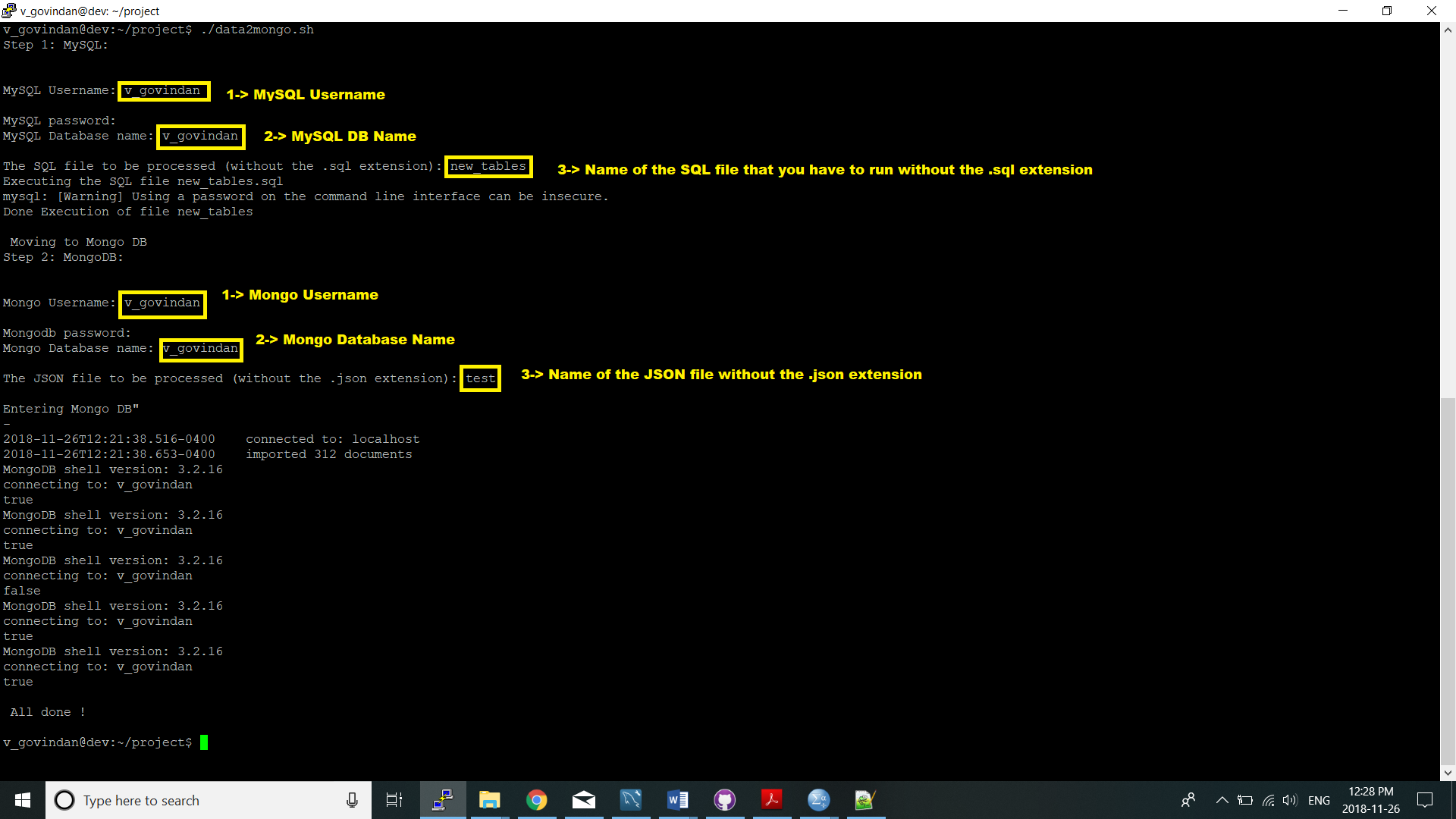


Figure Data2mongo script working

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.

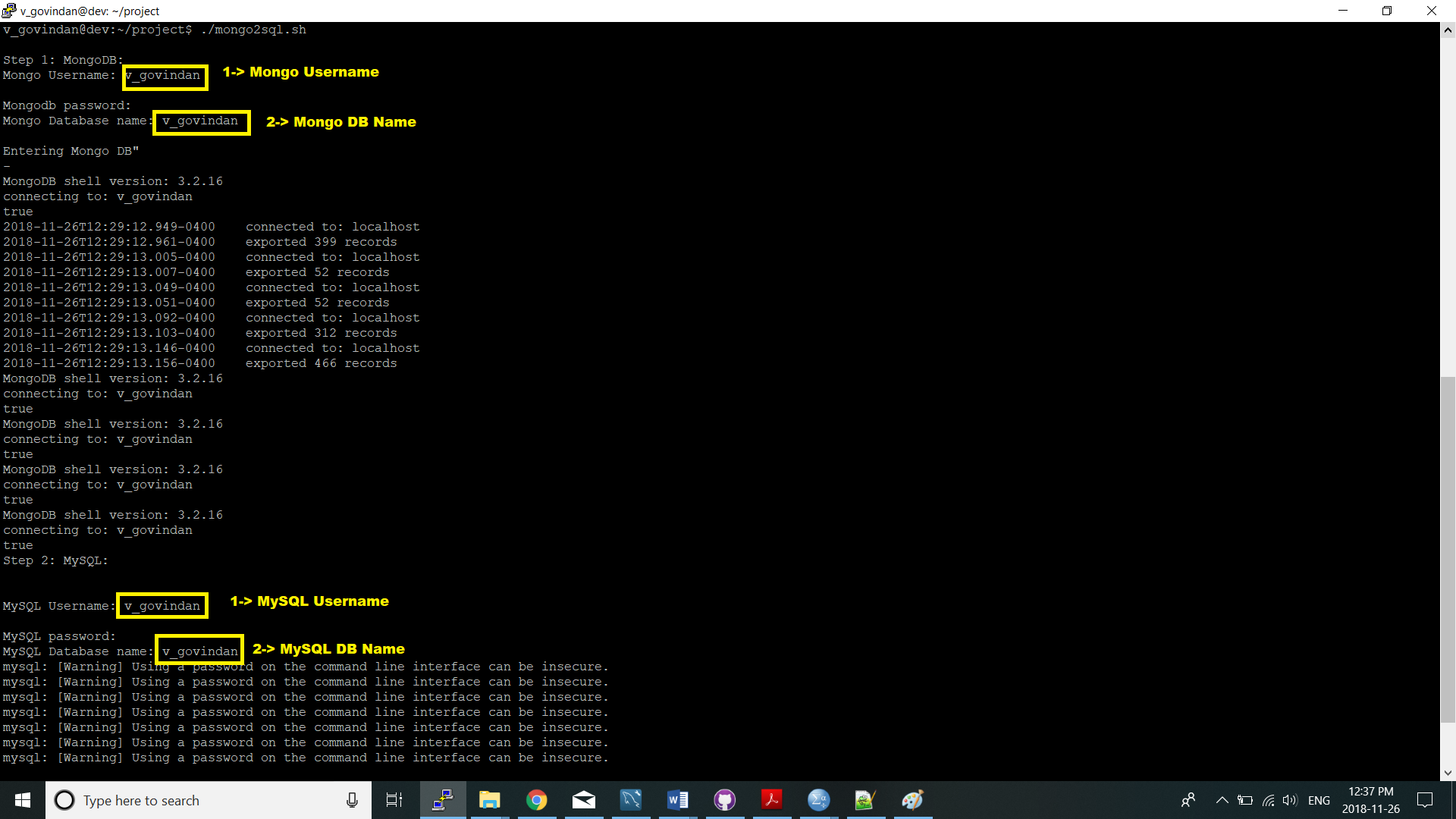


Figure mongo2sql script workflow

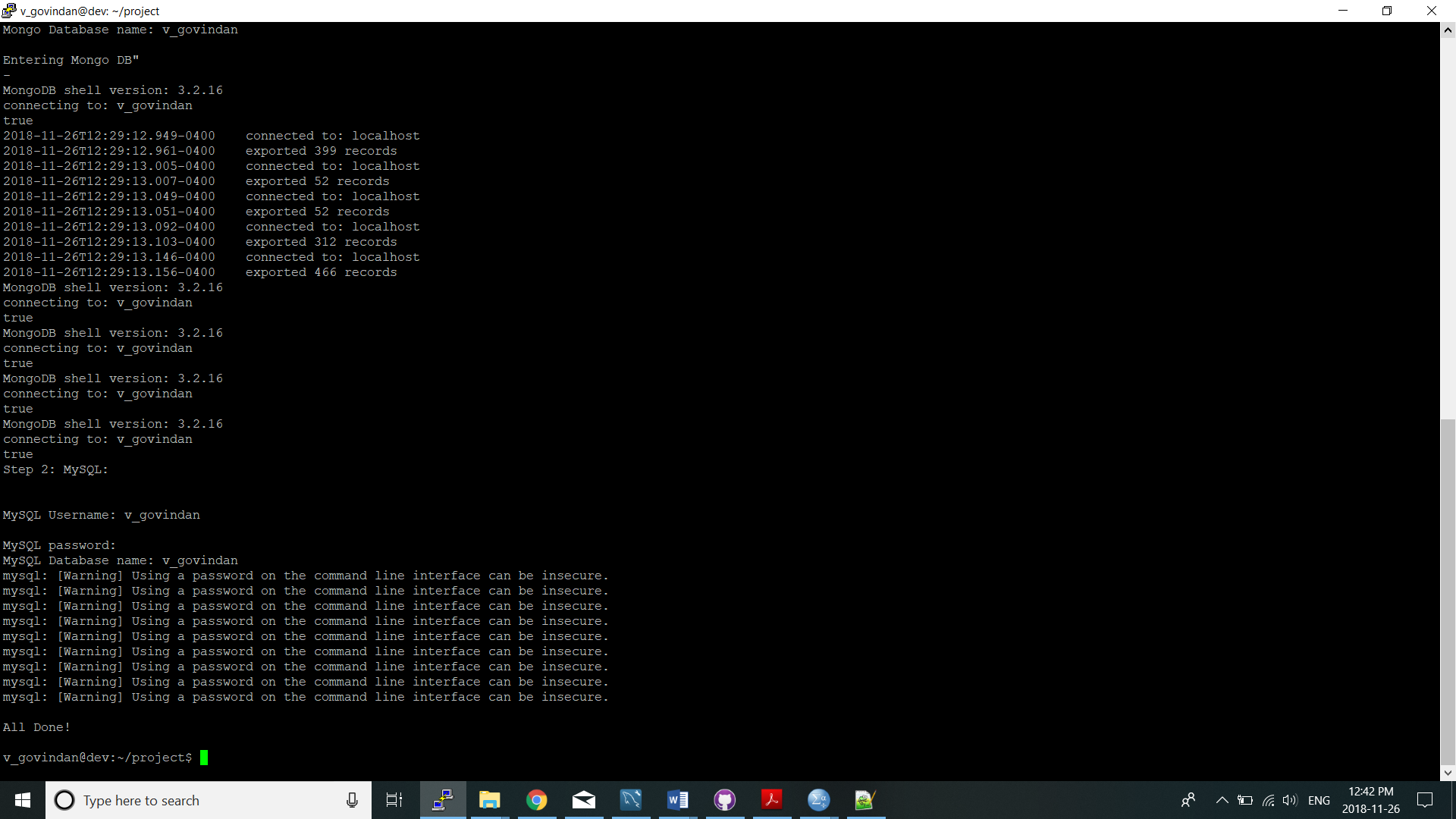


Figure mongo2sql script workflow Continue

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

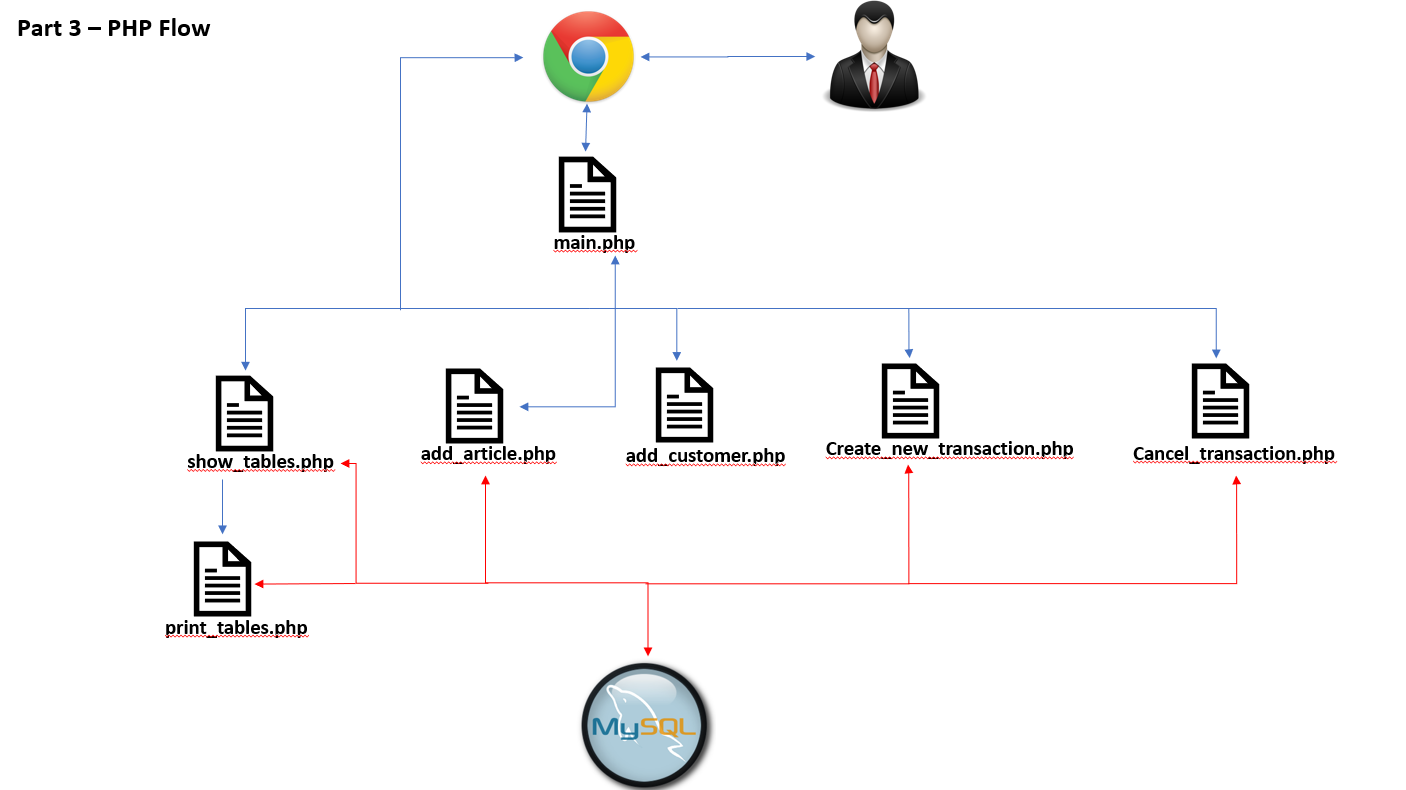


Figure Php Application workflow

## 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.

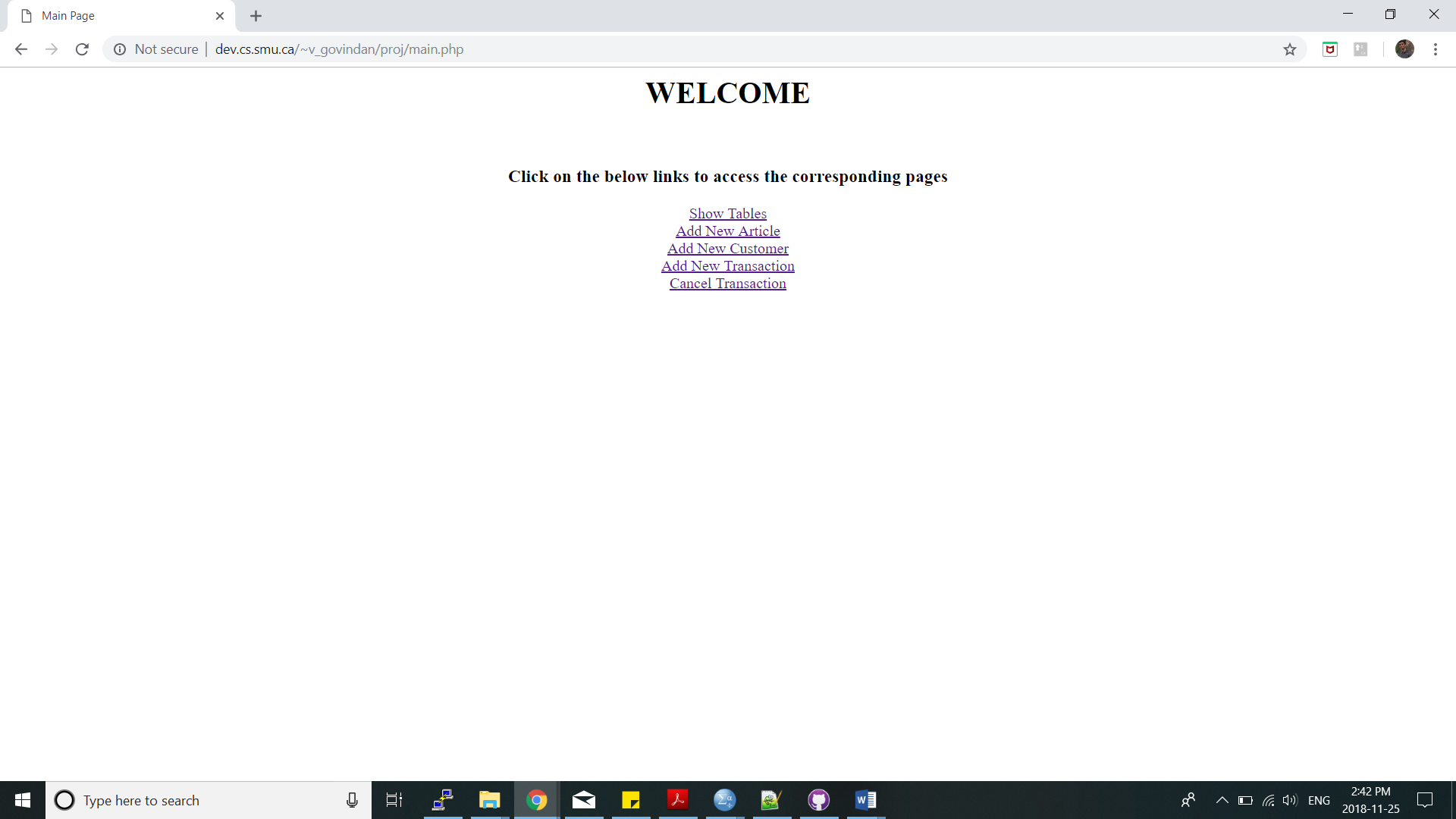


Figure Main page

## 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.

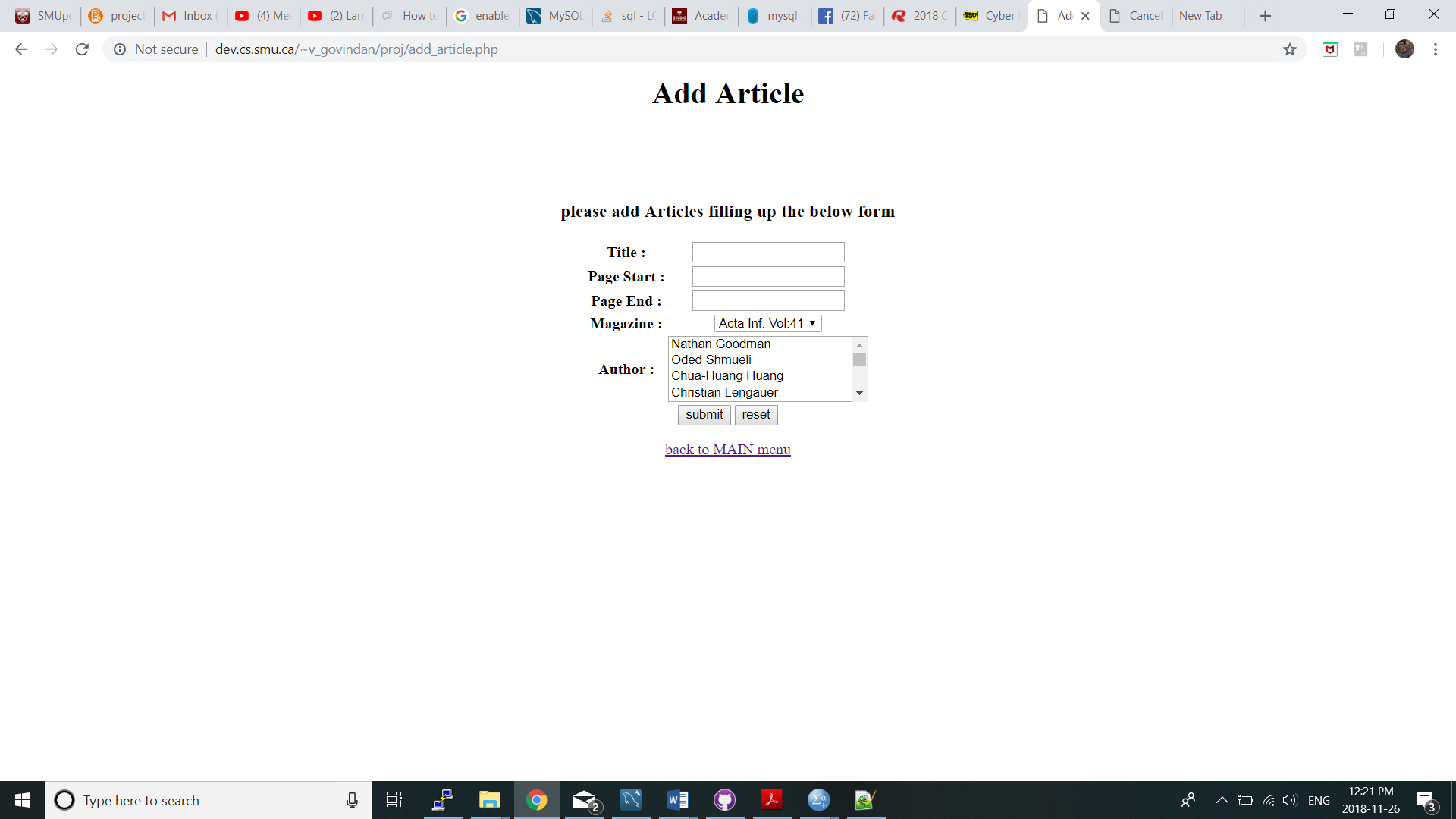


Figure Add Article page

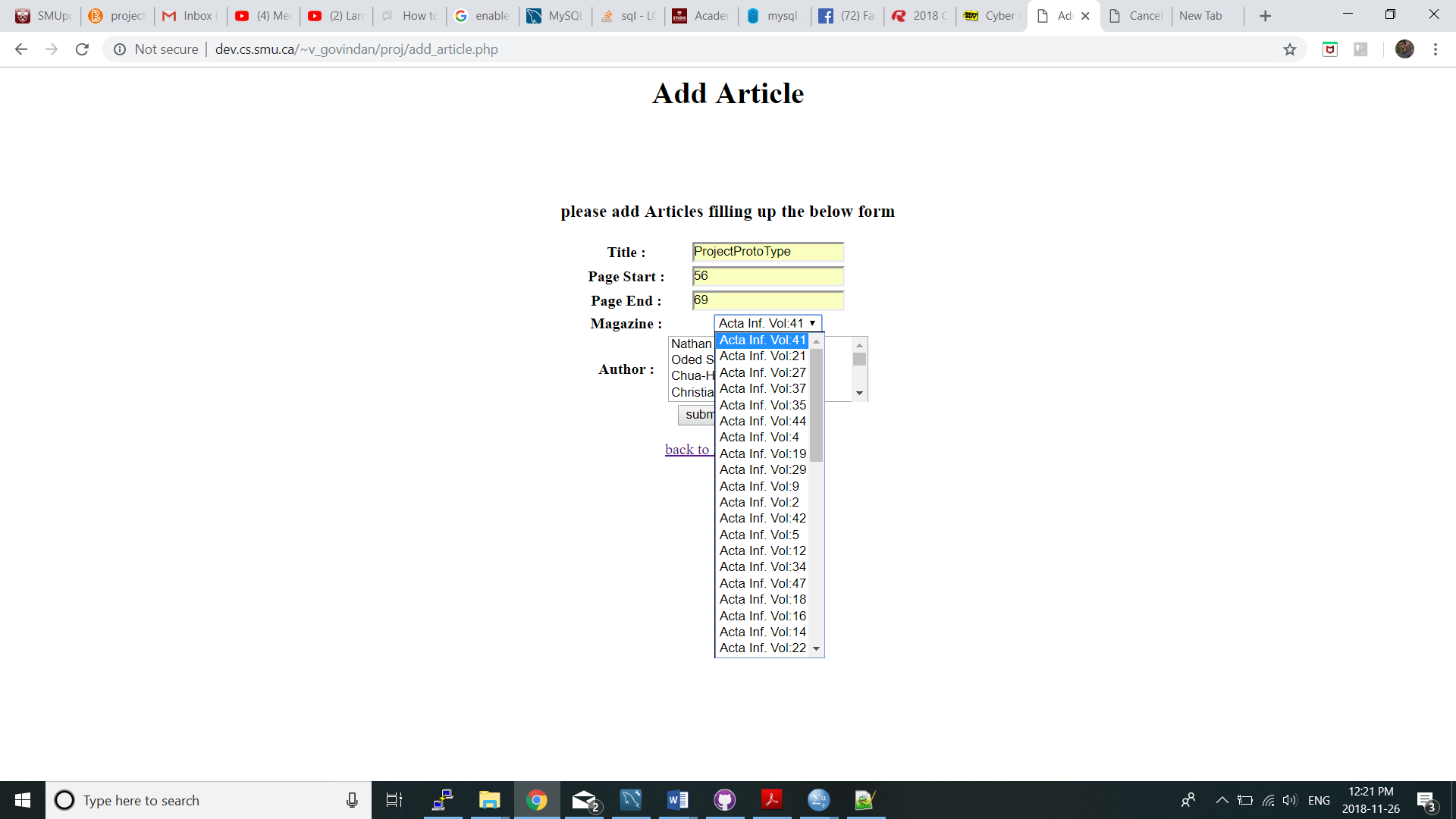


Figure Add Article Magazine selection

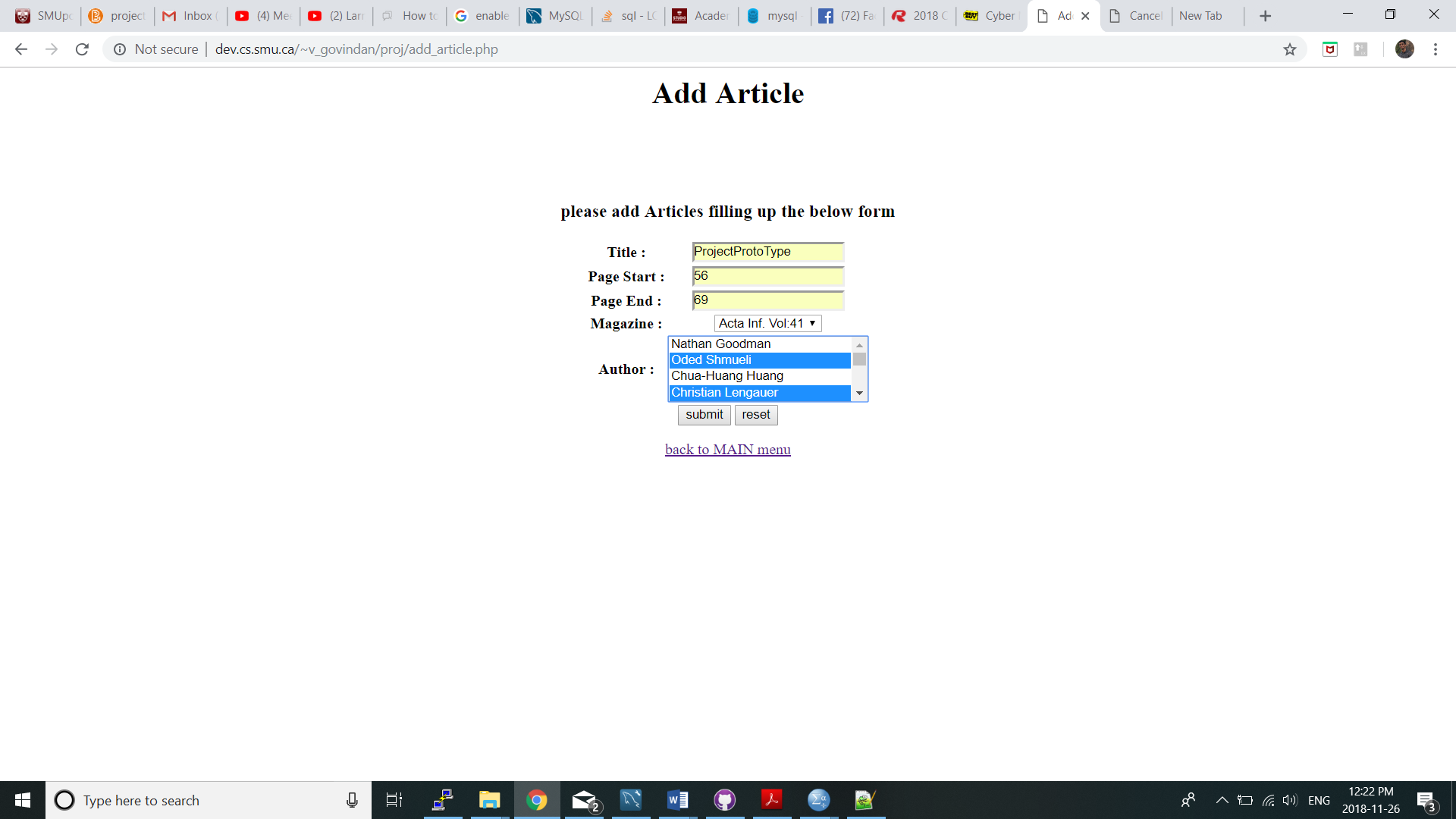


Figure Add Article Author selection

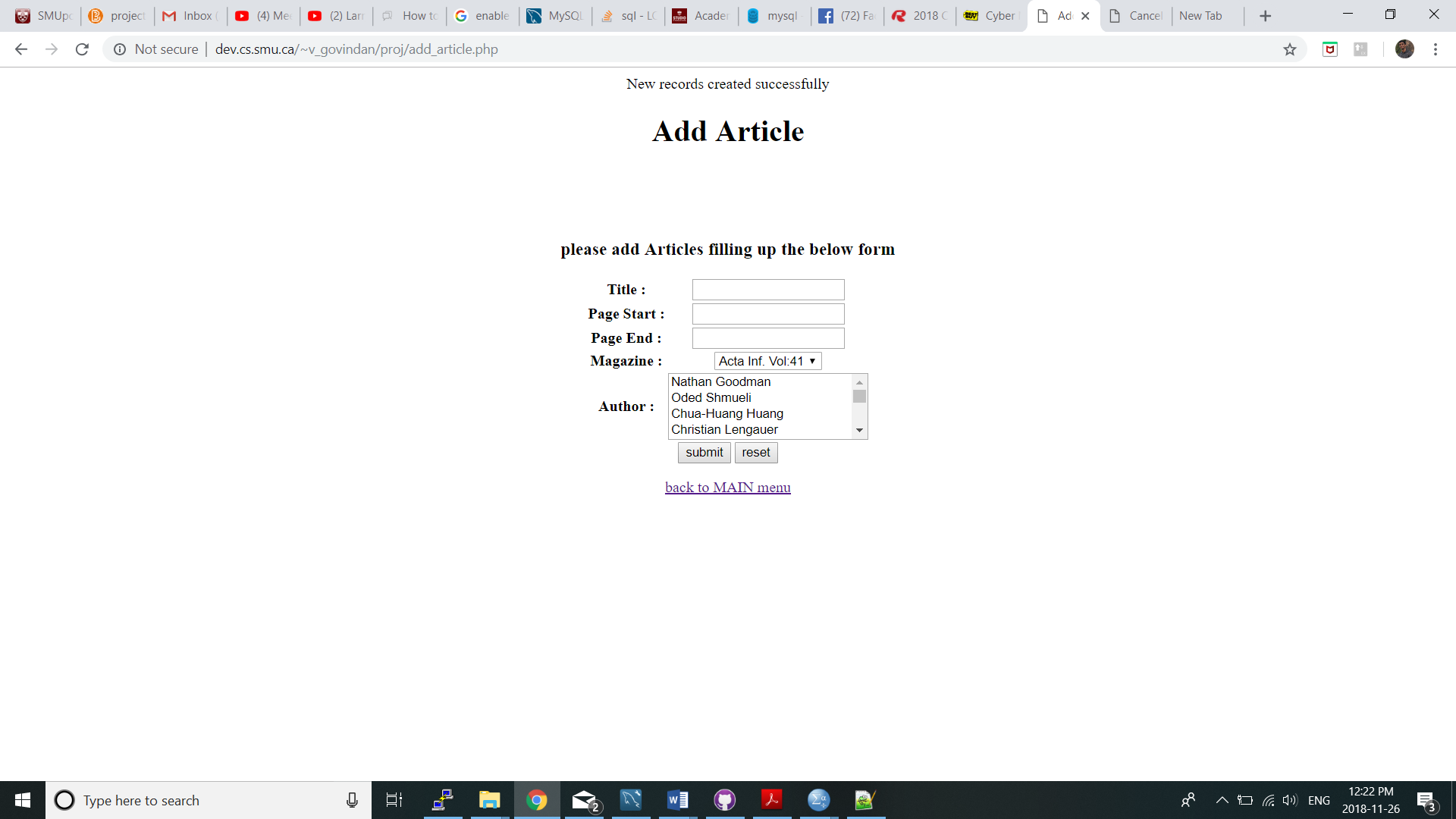


Figure Add Article Successful

## 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.

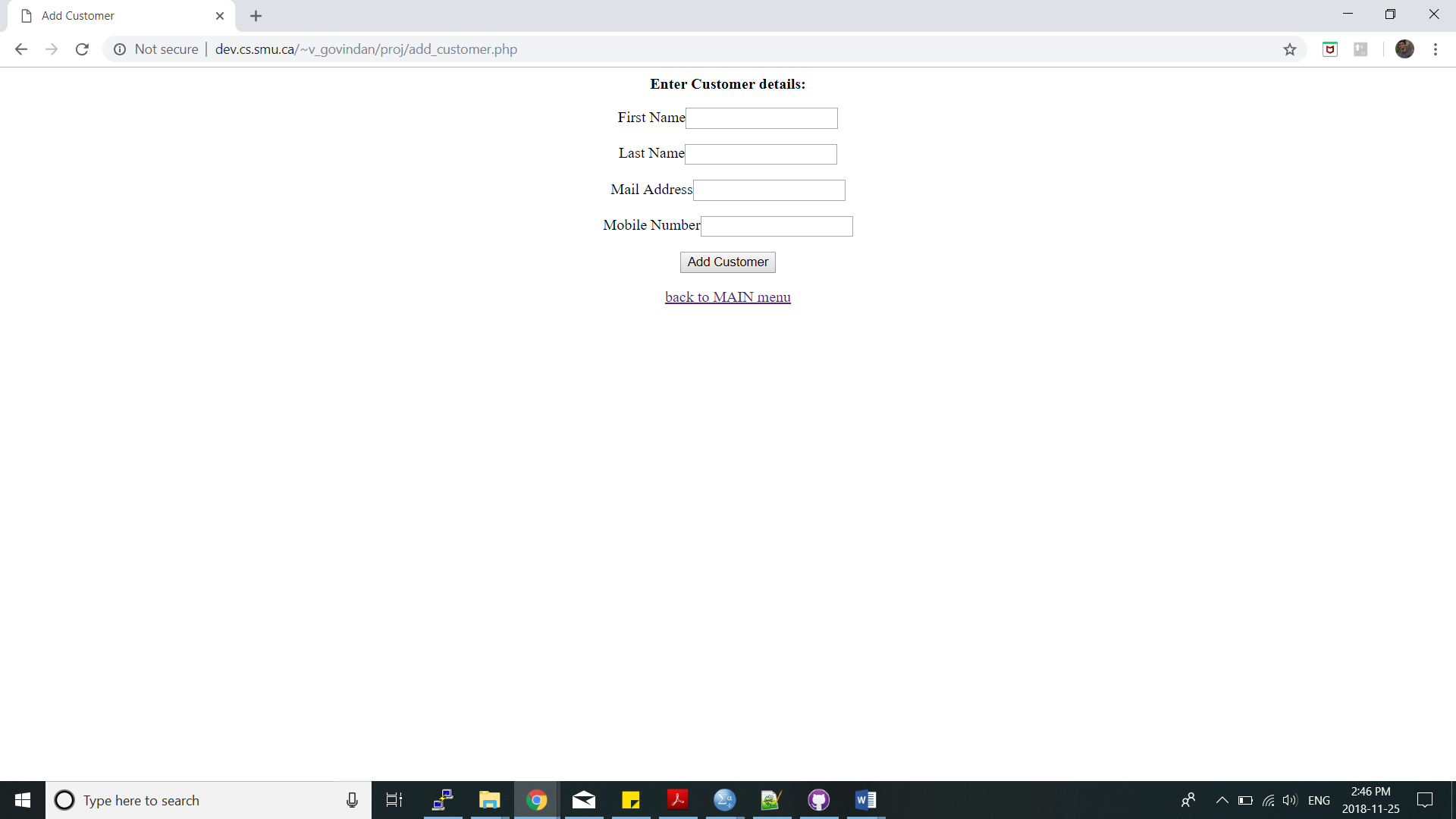


Figure Add new Customer

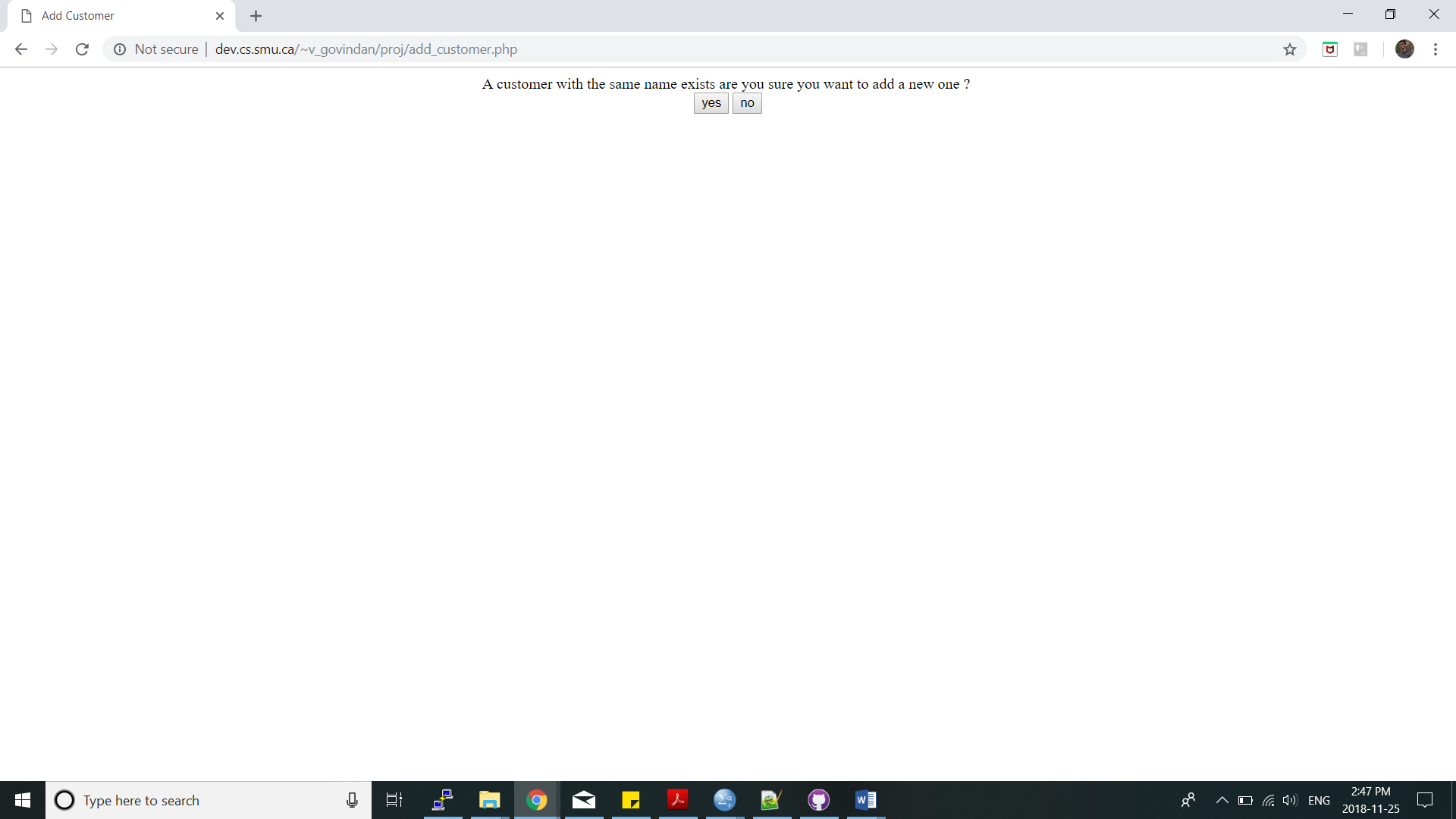


Figure Add new Customer duplicate entry Confirmation

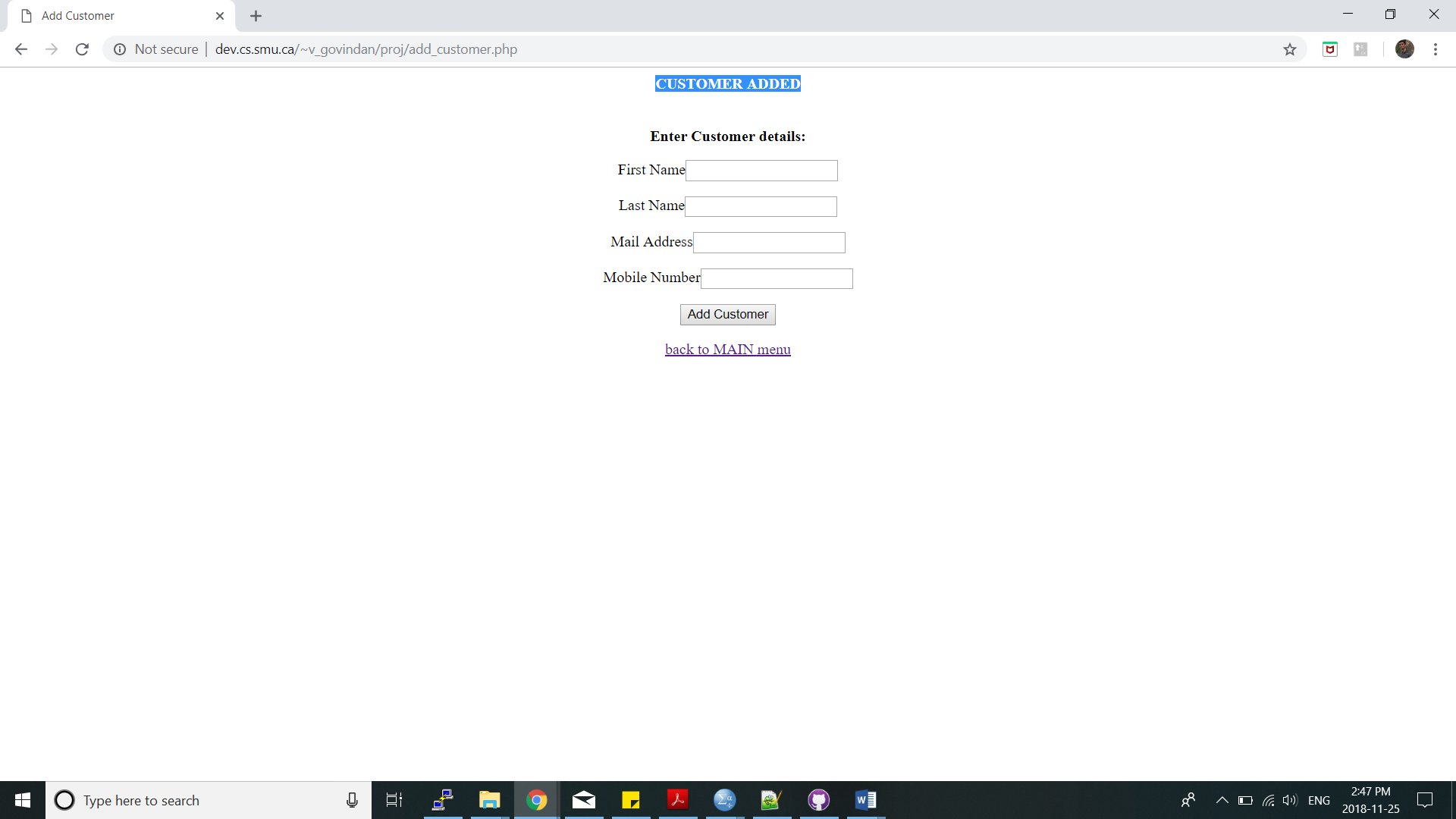


Figure Add new Customer Successful

## 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.

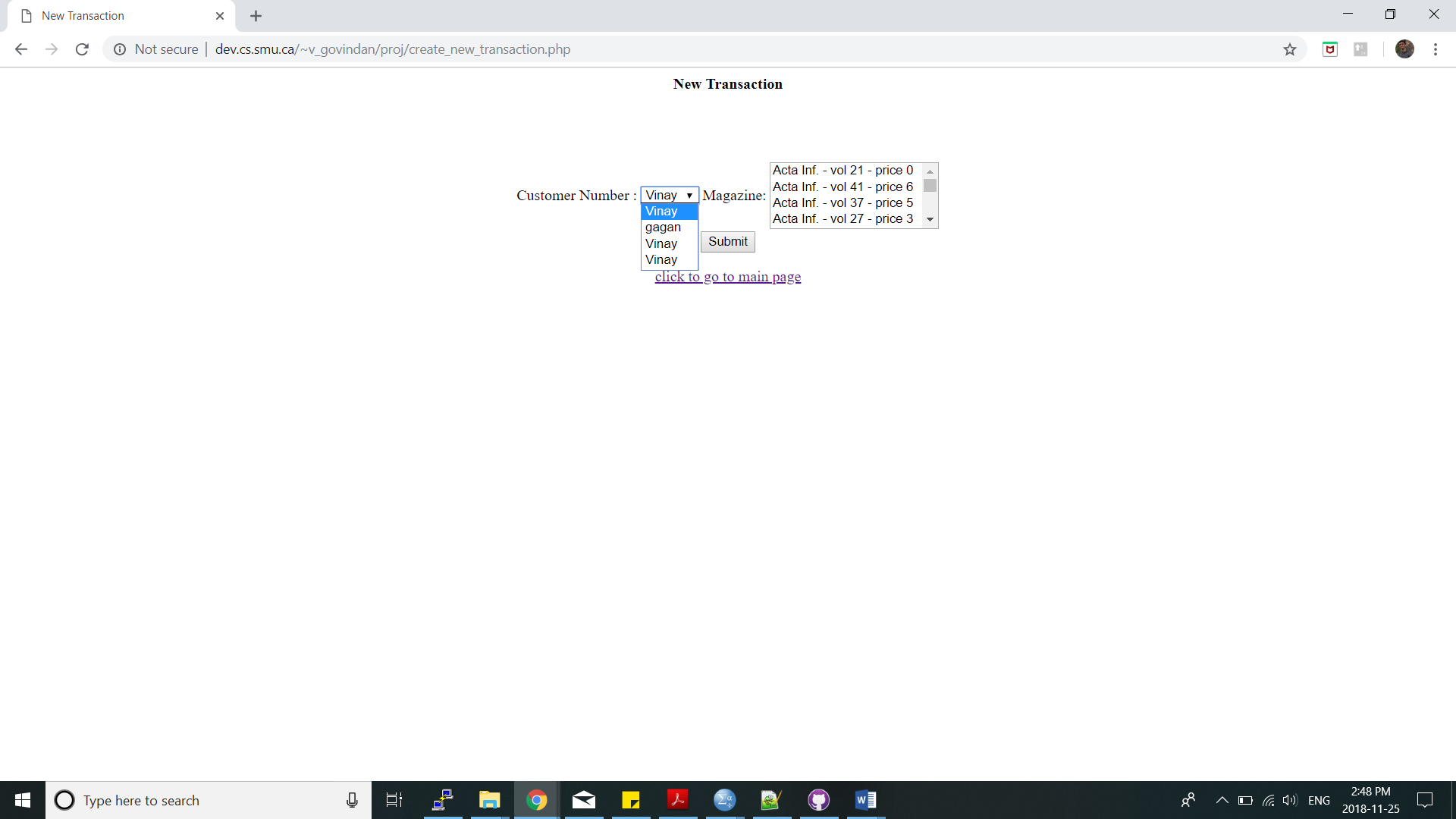


Figure Add new Transaction

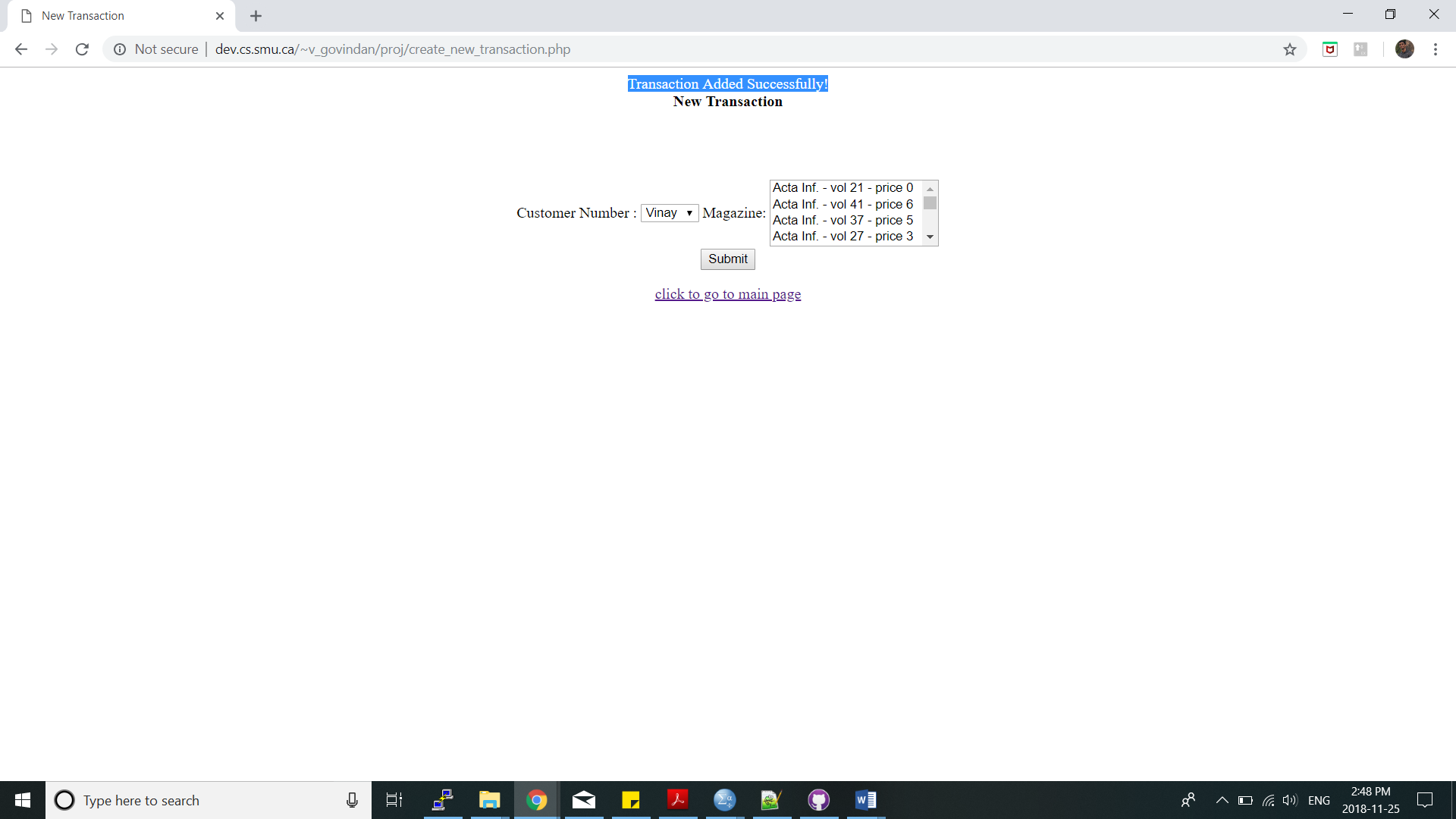


Figure Add new Transaction Successful

## 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.

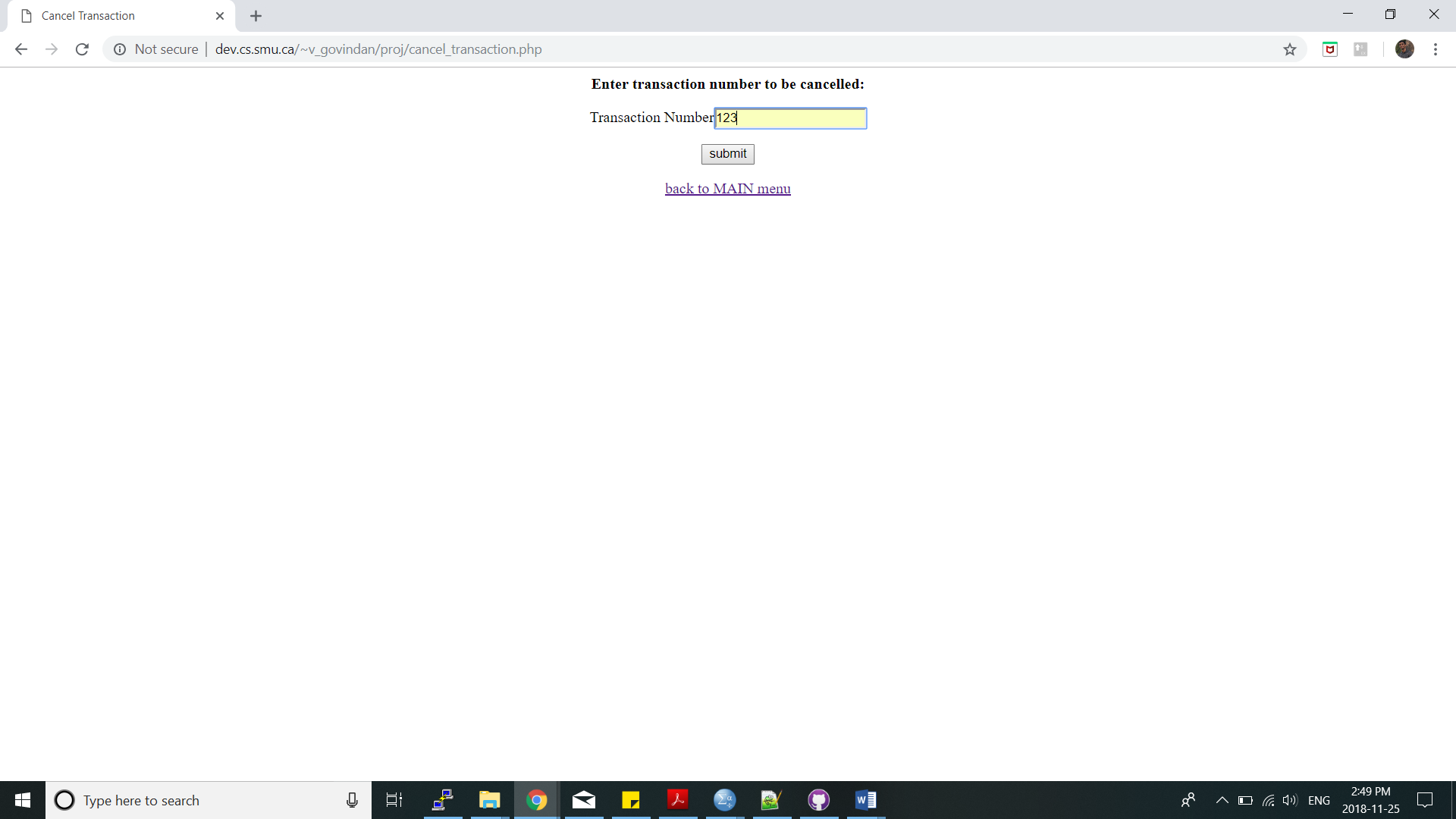


Figure Cancel Transaction

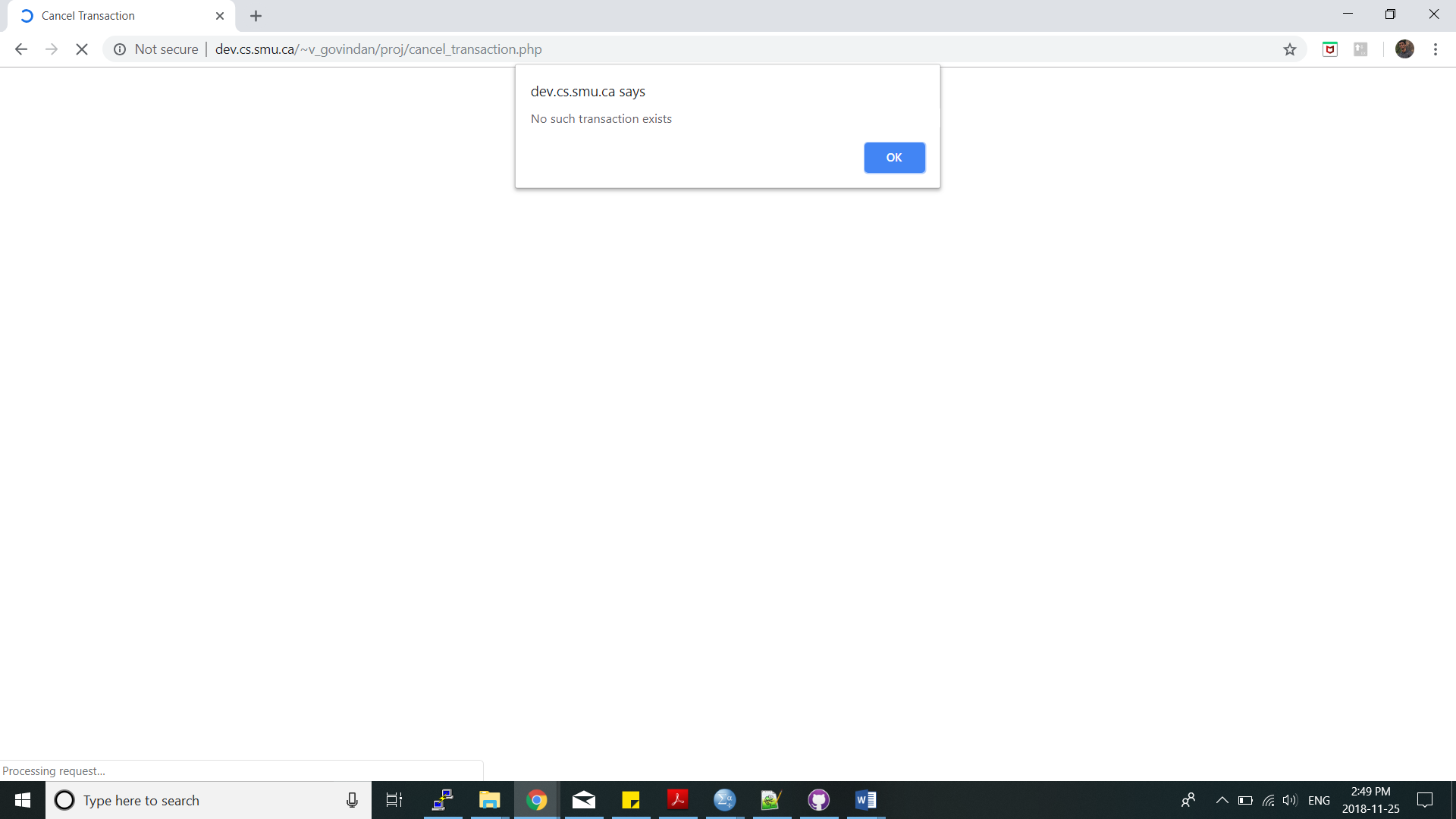


Figure Cancel Transaction Invalid ID Warning

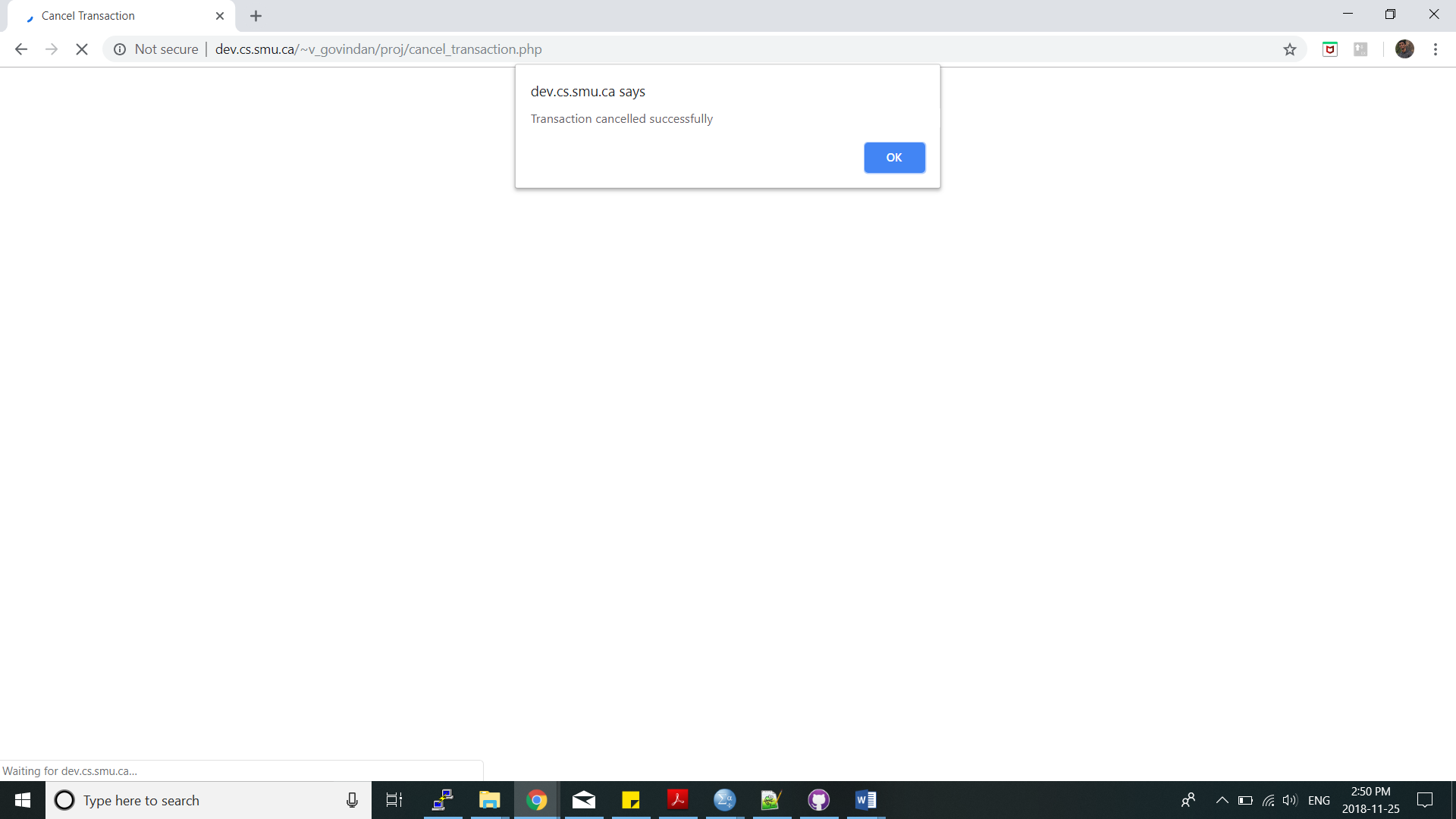


Figure Cancel Transaction Successful

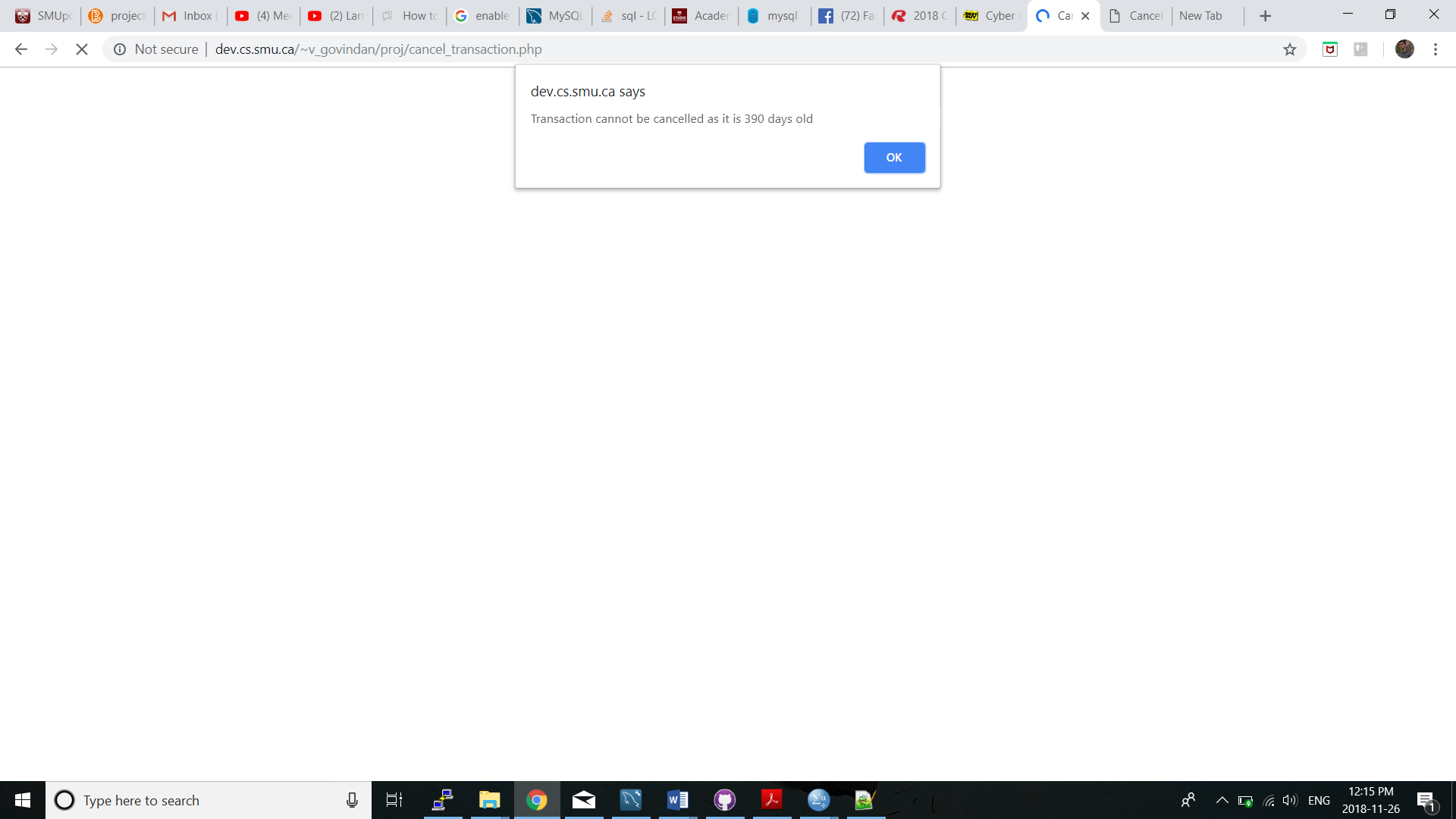


Figure Cancel Transaction Error if transaction is older than 30 days

## 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.

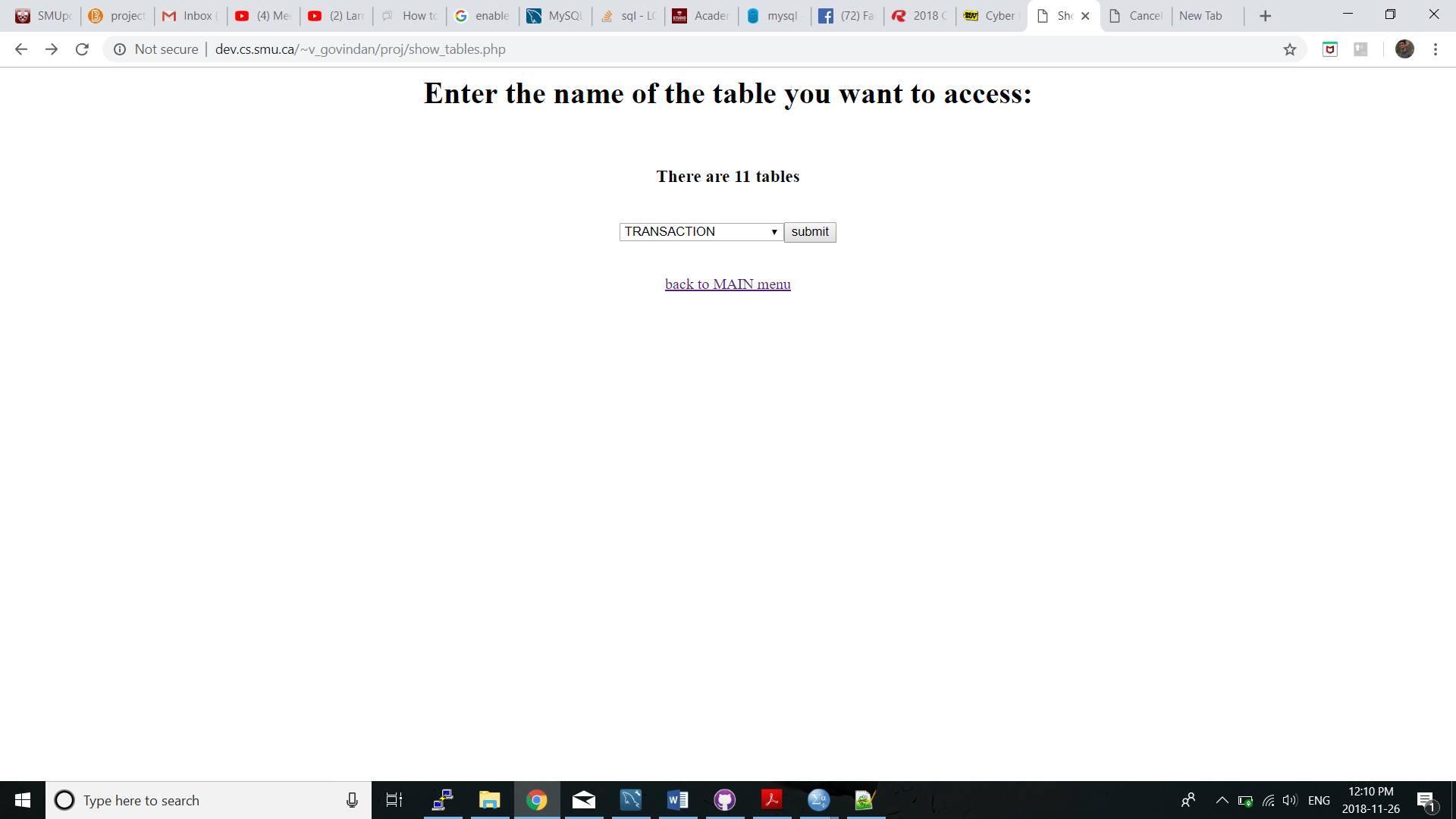


Figure Show Tables

## print\_tables.php:

Displaysdata of the selected tables.

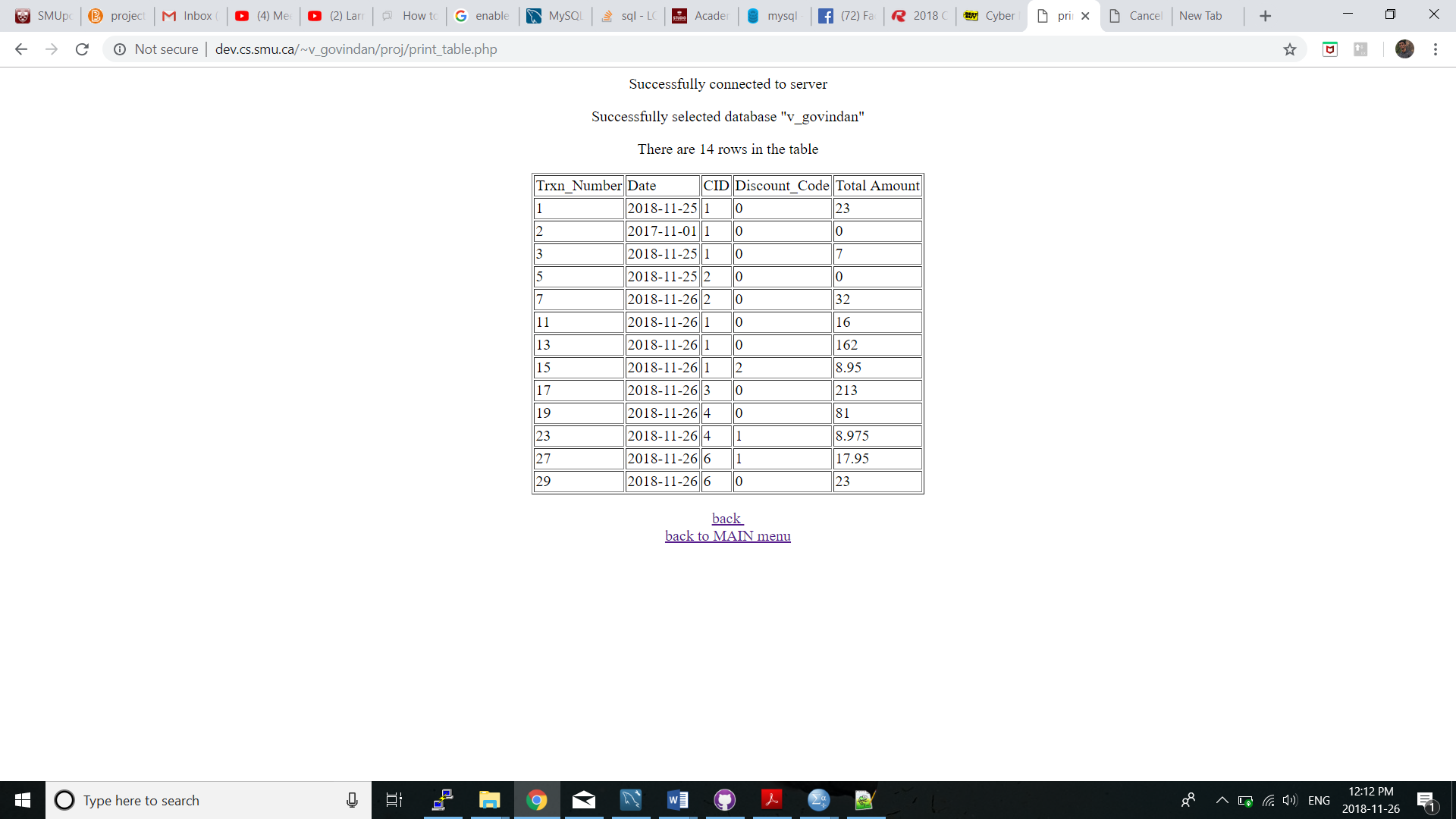


Figure Print Transaction table

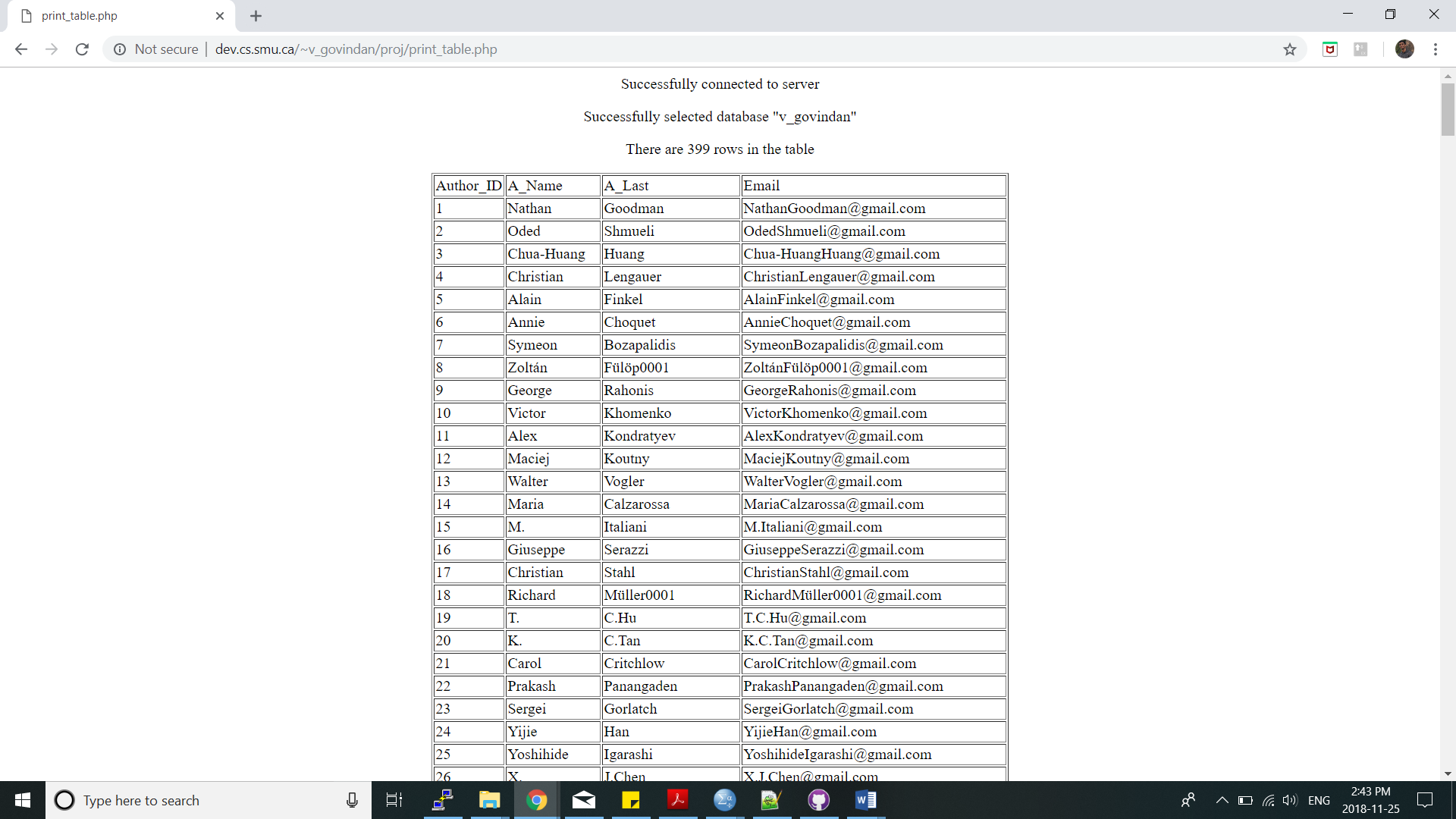


Figure Print Author Table

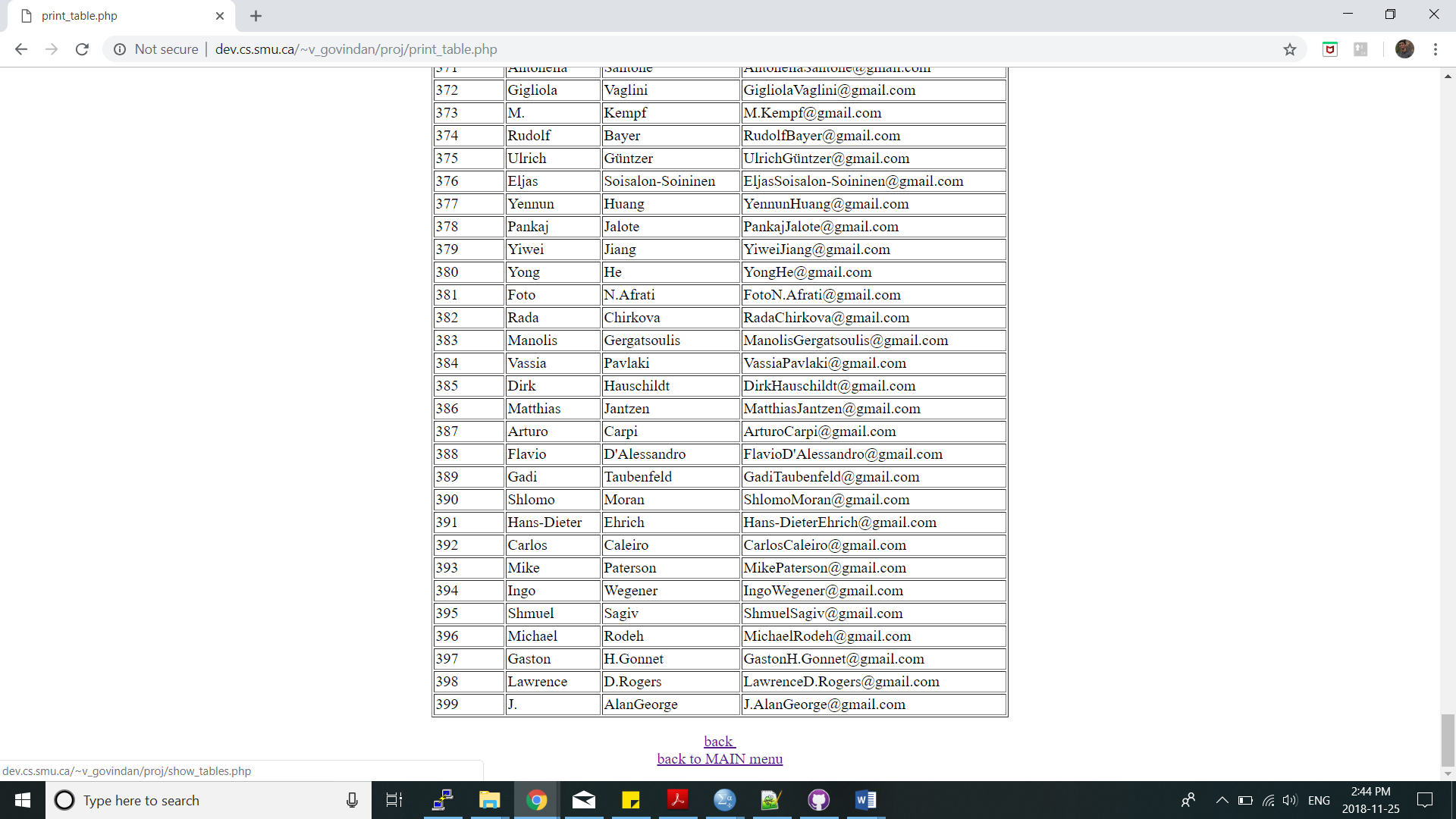


Figure Print Author Table Continue

# References:

1. [w3schools](https://www.w3schools.com/)
2. [stackoverflow](https://stackoverflow.com/)
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