REINARD SERFONTEIN

ITRW 324 PROJECT

Table of Contents:

[1. Introduction: 1](#_Toc26365587)

[2. Functionality of the system: 1](#_Toc26365588)

[2.1 Functionality of personal contacts: 1](#_Toc26365589)

[2.1.1 Login form: 1](#_Toc26365590)

[2.1.2 Sign up form: 1](#_Toc26365591)

[2.1.3 Choosing from: 1](#_Toc26365593)

[2.1.4 Personal Contact Details enter Form: 1](#_Toc26365594)

[2.1.5 Display all personal contacts: 1](#_Toc26365595)

[2.2 Functionality of business contacts: 1](#_Toc26365596)

[2.2.1 Entering data form for business contact: 1](#_Toc26365597)

[3. Problems faced during the project: 1](#_Toc26365598)

[4. Conclusion: 1](#_Toc26365599)

[5. Appendix A: 1](#_Toc26365600)

[6. Reference list: 1](#_Toc26365601)

# Introduction:

This second opportunity project is about creating a website or application that a person can use to keep track of users.

Allowing the functionality to insert, update, and delete of users.

A website was created to maintain and do accomplish all of this functions.

Where to start and what software and technology to use took some time to figure out. Having a good background of software that can be used to create a website and a server is helpful. Research and studying up on the subject, can just benefit you in the process of the project.

The software and technology that was used in this project is just HTML and ejs files for the front end of the project.

The coding of websites was done with basic HTML and CSS to give the websites some style and the server was created making use of MySQL workbench to be able to run the server from your local machine.

Backend of the website was created with node.js. This allows the website to establish a connection with the server to get and pass information between the website and the server. Vue is used to be the “middle man” between the front end and the server end.

Vue has a couple of functions namely:

Vue.js enables you extend HTML with other HTML attributes called directives**.**

Vue.js directives offer’s functionality to all of the HTML applications you are using.

Vue.js provides already built-in directives and multiple user defined directives.

Choosing to make use of HTML, node.js and Vue. At the same time using MySQL workbench to be used as my local host server.

# Functionality of the system:

## 2.1 Functionality of personal contacts:

The system was created by making use of a couple of different software and coding languages that will be explained later on.

The back end of the website was done making use of Node.js.

Node.js is:

* Open source server environment.
* It runs on various platforms (windows, Linux just to name a few).
* It makes use of java script on the server.

Node.js allows the website to be able to communicate to the Vue program that then retrieves the necessary data from the server that was created in MySQL Workbench as a local host server.

The server can be created and run on a virtual machine that can be created with Microsoft azure. This virtual machine makes it allot easier to be able to communicate with it from any place, not needing to install MySQL workbench on your local machine to be able to run the website.

The basic functionality to be able to connect to the server is done with a simple connection query that can be found in the code under the connect.js file.

This file has all of the code needed to establish the connection between the website, Vue and the MySQL server.

The file can be run in the command prompt window. By typing in node coonect.js the code will execute and establish the connection.

Once the connection is established the command prompt will display a message showing the server is running on port 3000.

By entering localhost:3000 in your browser the website will be displayed.

The first page the user will come across is the login page.

### 2.1.1 Login form:

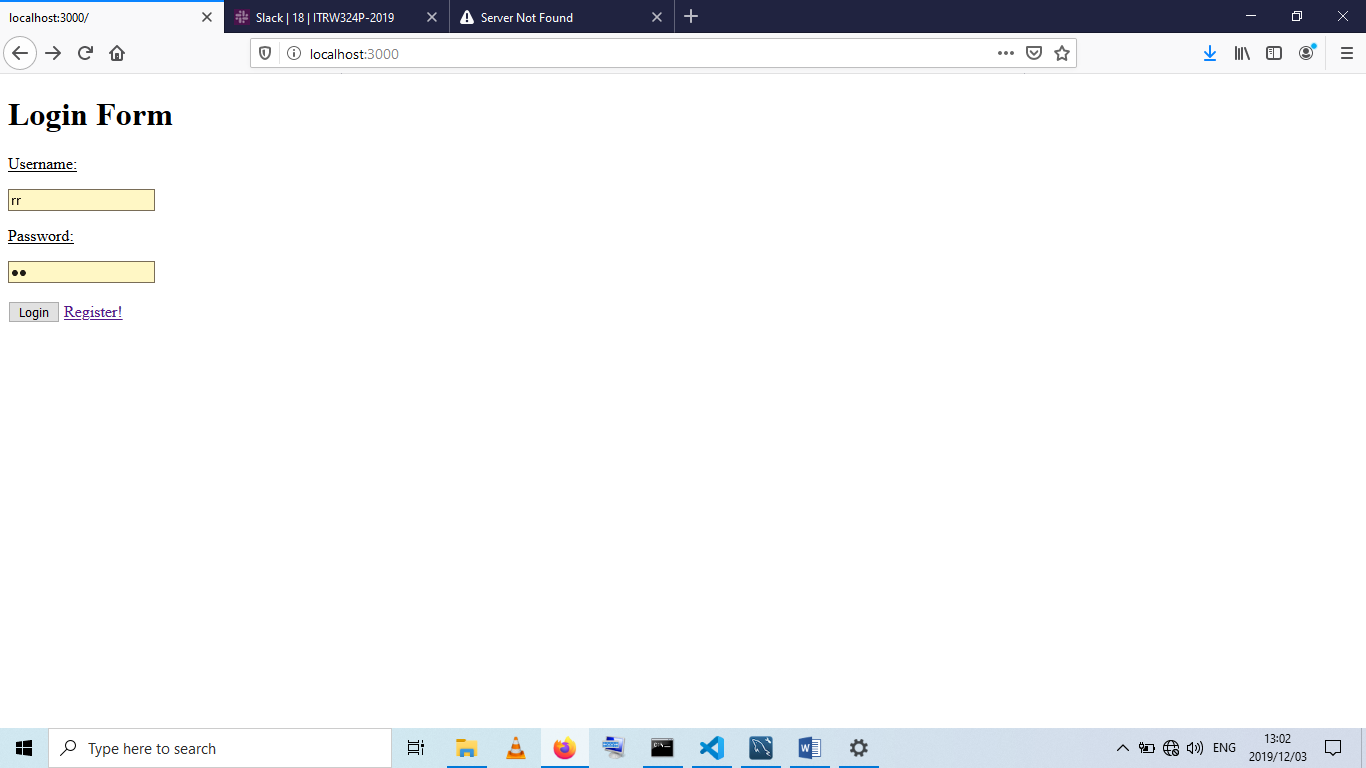


Figure 1: LOGIN FORM

This page prompts prompt’s the user to enter there details to be able to gain access to the system.

When the wrong credentials are entered the system will then render a page that shows a message, “Please enter the correct details”.

In the background the website will make use of SQL query’s that was written, to validate and make sure the info enters matches that of that is registered on the system and at the same time make use of the bcrypt package to decrypt the hashed password to allow the system to compare the password and make sure they are the same.

The query retrieves all of the necessary information from the database with a SELECT \* statement and then compares it to the data that was entered into the login page.

If the user is not registered on the system, the option is given to allow the user to register.

Once the user clicks on the register button the register page will come up.

The login page is used for both personal and business contacts.

### 2.1.2 Sign up form:

# 

Figure 2: SIGN-UP FORM

This page allows users to enter the necessary details and once they click on the create user the website will communicate with the database and create the user.

The page has certain functionality that will check if the user has entered information into all the fields. If a filed is left open or don’t have any information entered into it, a message will be shown next to the field saying that it is required to be filled in.

The query will execute with a SQL statement that INSERT username INTO the correct table.

With the inserting of the information into the database, the bcrypt package is installed to hash the password and to encrypt passwords to be able to keep users passwords safe.

If the user is successfully created the website will render a page that will display a message, “User successfully created”. The user can then go back to the login page and enter there details they have just created and then will be granted access to the system.

This way to gain access to the system is used, because keeping track of who is login in and making changes is important. This way when a specific user logs in, they will only have access to certain things on the system. A personal contact user can only see personal contacts details, and a business contact can only see business contact information.

### 2.1.3 Choosing from:

Figure 3:SELECTION PAGE

Once the user gains access to the system, the page will be displayed where the user can those to enter a business contact or a personal contact.

With this page the user is given the option to logout of the system at any time they are making use of the website.

I just made use of a normal href link to manage navigation between the pages.

### 2.1.4 Personal Contact Details enter Form:

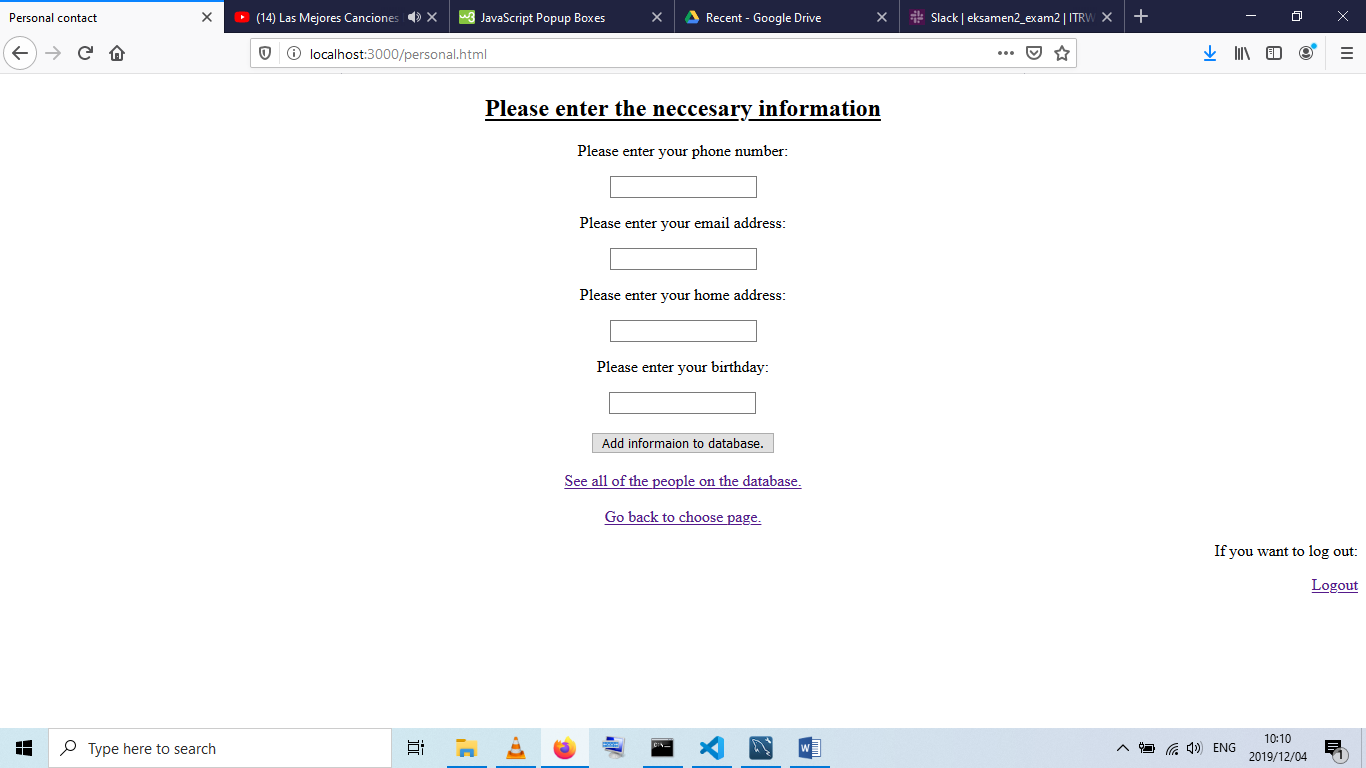
Once the user clicks on the personal contact button the page will appear where the user can enter their business details that they want to store on the system.

Figure 4: ENTER PERSONAL DETAILS

This page also makes use of SQL query’s to be able to communicated with the server.

The queries INSERT the information into the corresponding table on the server.

The structure of the query that inserts the information into the corresponding table on the server is:

"INSERT INTO personal (phonenumber, email, address, birthday) VALUES (?,?,?,?)"

As the same as the register page, this page will verify that all of the fields have information entered into them. If a filed is left black a message will appear next to the field saying that it is required.

The website will also verify that the information entered is in the correct format. For example, the phone number must be 10 digits long and the email address must be a valid email address before the information can be stored onto the server.

Once the user presses the button to add the information to the server the fields will then clear and be ready for the next entry.

The options to see all the records in the system is also available along with the option to go back to the page where the user has a choice of personal or business contact.

### 2.1.5 Display all personal contacts:

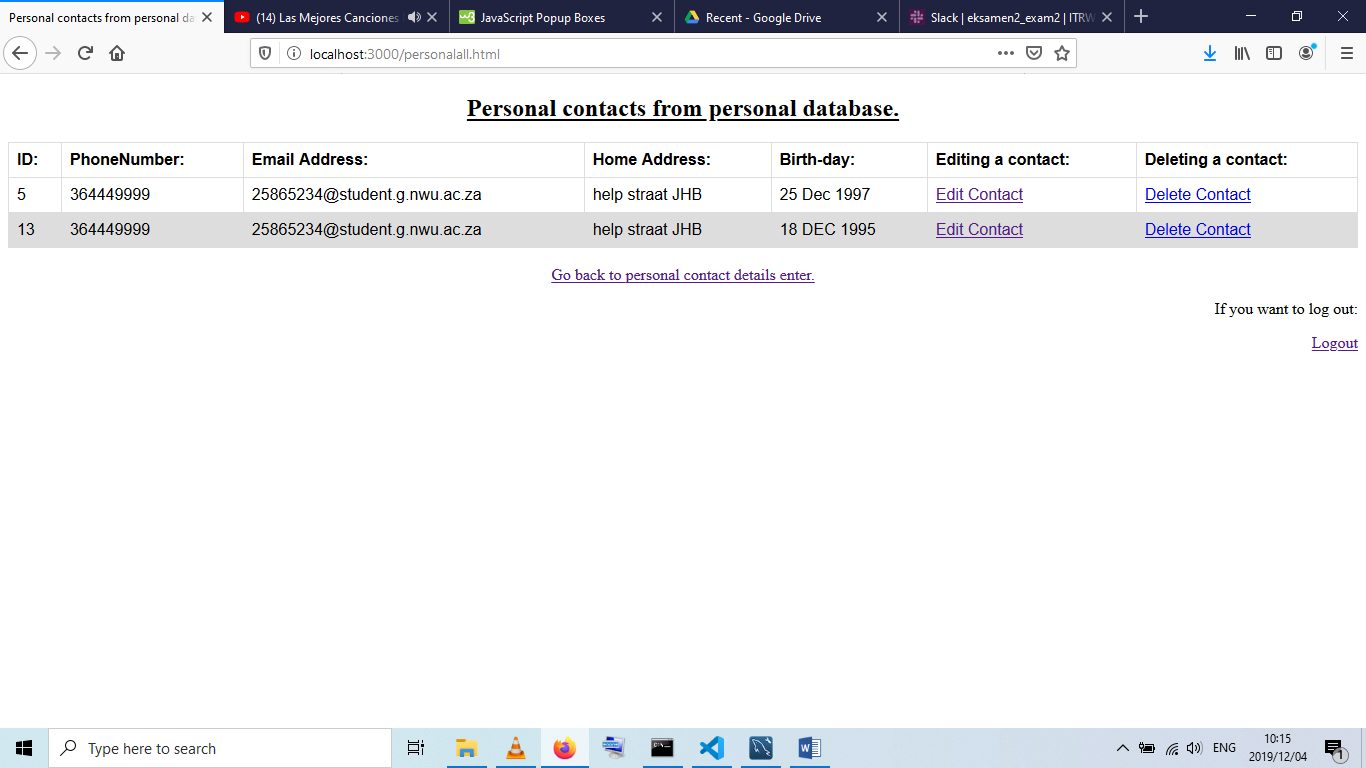
If the user chooses to see all of the records that is stored on the system and clicks on the button, the website will render a page containing all the necessary information.

Figure 5: TABLE SHOWING ALL PERSONAL CONTACTS

The table on this page is being completed with all the records that is stored on the server under the personal table.

Firstly, the website has to check if there are any records on the server. If there is records the website will execute the query to fetch all the records from the selected table.

The queries structure is as follows:

* SELECT \* FROM personal.

More functionality was added to the website. The options to edit or delete contacts that is stored on the server.

The buttons have the ability to delete records from the database with corresponding user ID that is in the table shown on the website.

This page allows the user to make changes and delete user profiles that is stored on the server.

No matter what option the user chooses the website will show a prompt confirmation message with both of the options. It asks the user, just to make sure the user is knowing what they are doing, “Are you sure you want to delete or edit the following contact?”.

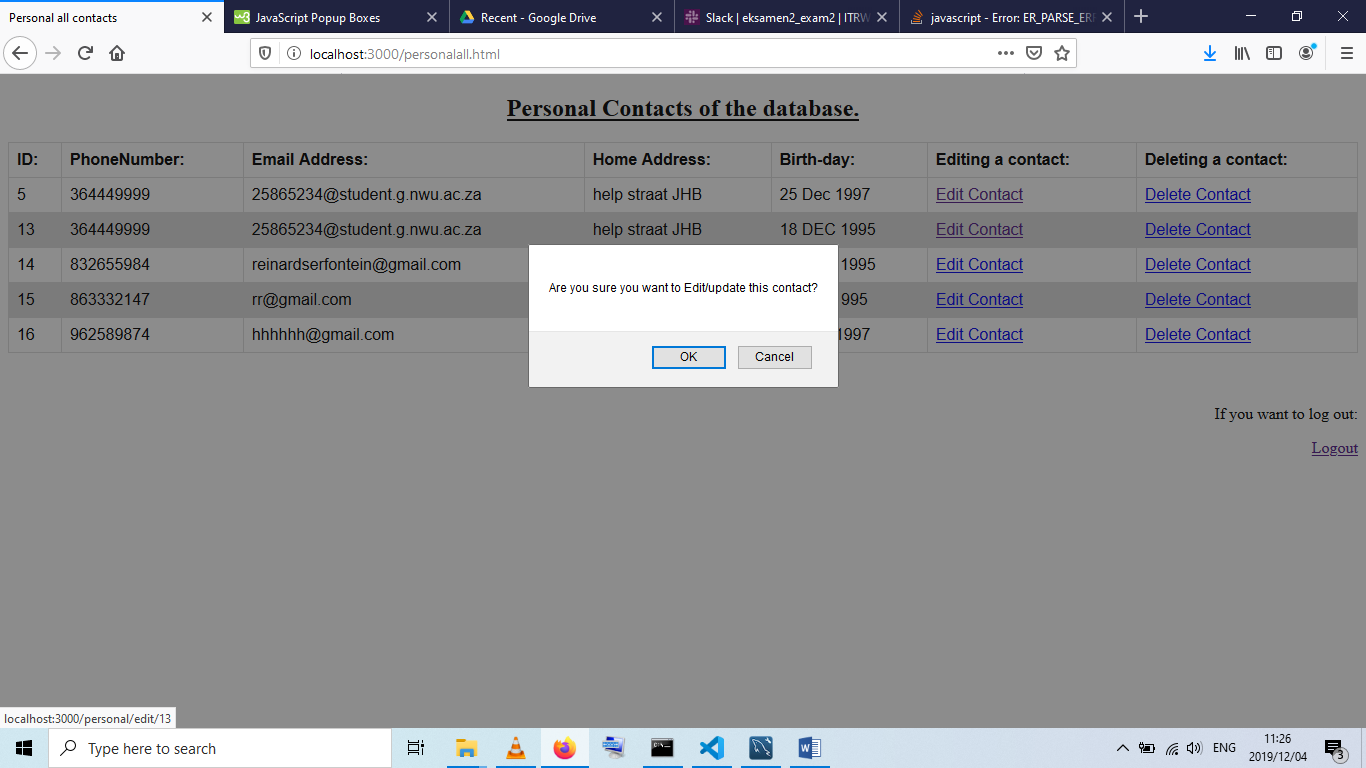


Figure 6:MESSAGE THAT PROMPTS CONFIRMATION

This prompt message is just there to add extra safety for if the user didn’t mean to click on the delete button. So the user has the option if it was an accident they can say cancel and the website won’t do anything.

If the user wishes to edit a profile on the system, the following page will be displayed.

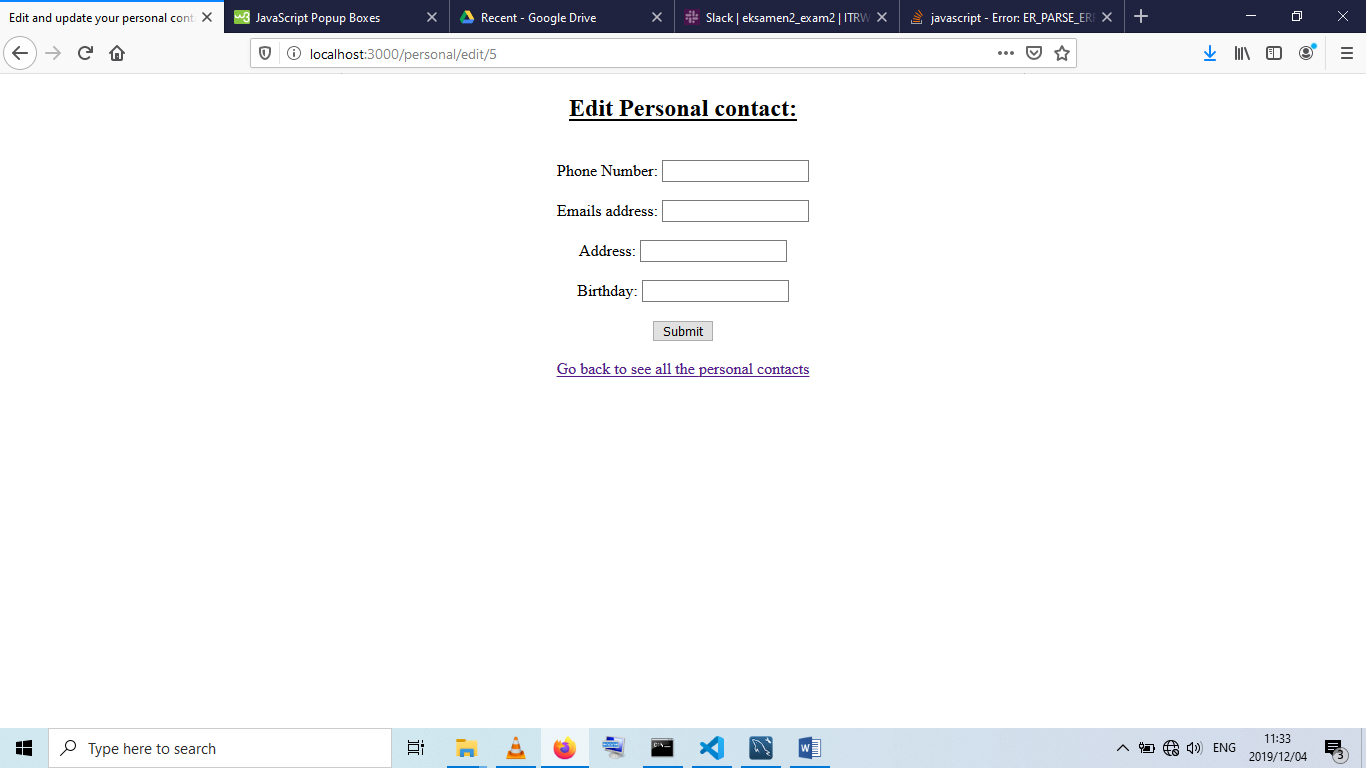


Figure 7: FORM FOR EDITING PERSONAL CONTACTS

Changes that is made to the profiles are executed by a SQL query that UPDATE the necessary profile according to the userID that was selected.

The SQL query is structured like this:

UPDATE personal SET phonenumber='"+request.body.phonenumber+"',email = '"+request.body.email+"', address = '"+request.body.address+"', birthday = '"+request.body.birthday+"' WHERE id = "+userId;

To make sure that the changes has been made, returning back to the command prompt a message will be displayed for the user to see confirmation, or the user can return to the personalall page where all of the personal contacts is displayed.

The page will then get the profiles on the server and display them in a table format.

The user also has the option to delete a profile from the server. This can be done by clicking on the delete a profile button.

The following page will be displayed.

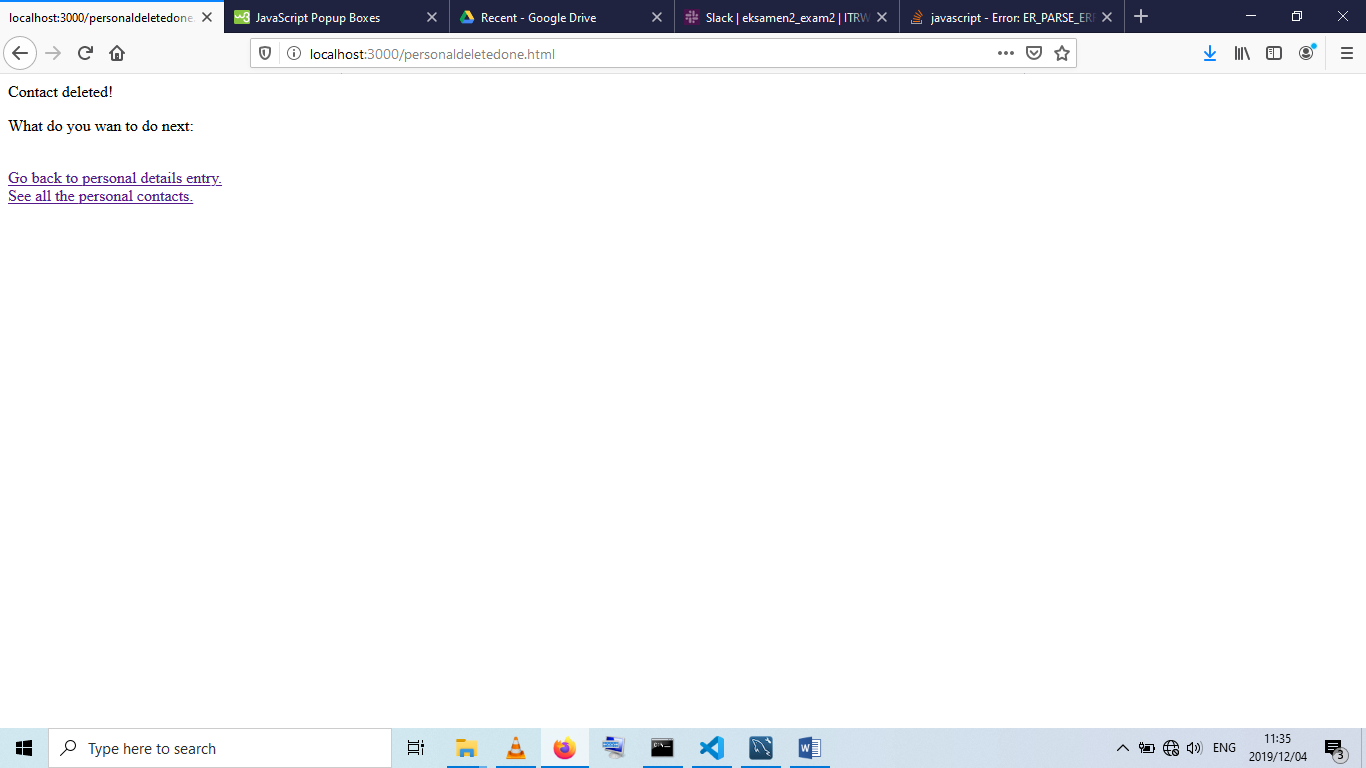


Figure 8: CONFIMATION MESSAGE OF CONTACT DELETED

When a user clicks on a contact in the table and then click on the delete button, the corresponding contact with the right user id will then be deleted from the system.

The SQL query will then run and DELETE the profile that was selected from the server’s database.

To be able to see if the contact is deleted the user can look in their command prompt and click on the button to display all of the contacts in the system.

The same functionality is applied to be able to achieve the same results for the business contacts.

## 2.2 Functionality of business contacts:

The login form and the signup forms are both the same for the personal contact and the business contacts.

The only difference comes in at the entering information that is going to be stored on the server.

### 2.2.1 Entering data form for business contact:

As you can see there is a slight difference in the two forms.

The personal form has a phone number, email address, birthday and a home address where the business contact has a phone number, email address 1, email address 2, home address 1, home address 2 and a VAT number.

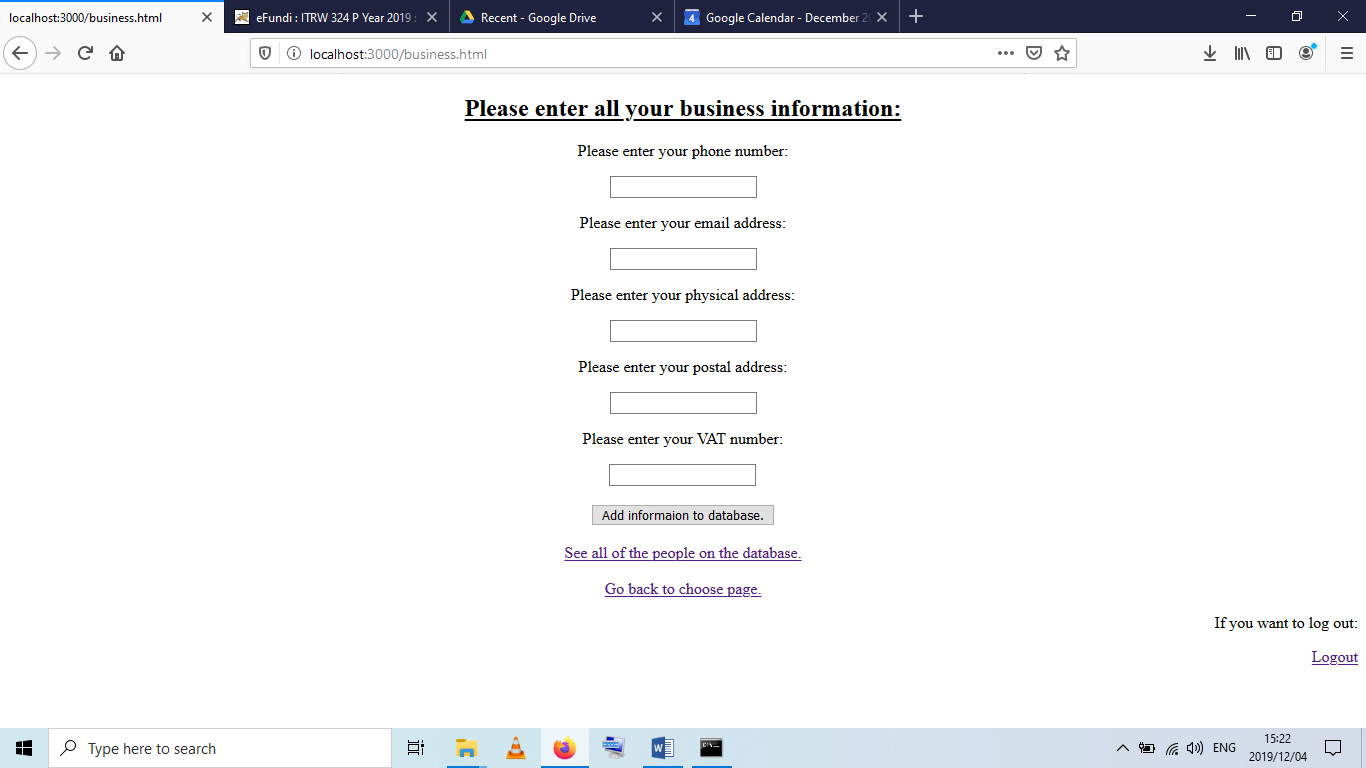


Figure 9: BUSINESS CONTACT INFROMATION INSERT

The displaying of contacts is done the same way as with the personal contacts.

The update and the delete buttons work in the same way as the buttons work for the personal contacts.

# Problems faced during the project:

The one main problem that came up was how to place the information that is being pulled from the server onto a website.

The solution was found when making use of the internet.

% signs where made use of to be able to tell the website that the information that is pulled from the server must be place into the table cell.

Another problem that came up was the correct use of the SQL queries.

The query needed to be the same spelling and the same caps otherwise the query won’t work and an error message will be displayed.

Trying to encrypt the password brought up a lot of errors. When trying to install the npm package for bcrypt, node.js said there was an error installing the packages.

So I created a package.json file to be able to install this package.

# Conclusion:

Learning a new language like node.js and Vue showed me how the website system actually works.

Having your client side. Having your “middle man” that is your Vue program. Then having your back and server side.

Learning that the client side sends a HTTP request to the Vue program. The Vue program gets the HTTP request and receives the required data from the server.

The server then sends back a JSON file that Vue can use for multiple things. As discussed above.

Vue then takes the data and the JSON file that was received from the server, and rearranges it so that it can be sent to the client side to be placed onto the website that the user can see.

This proses will be done every time the user searches for a website they want to look at.

# Appendix A:

[Figure 1: LOGIN FORM 1](file:///C:\Users\User\Desktop\ITRW324%20FINAL%20PROJECT.docx#_Toc26365663)

[Figure 2: SIGN-UP FORM 1](file:///C:\Users\User\Desktop\ITRW324%20FINAL%20PROJECT.docx#_Toc26365664)

[Figure 3:SELECTION PAGE 1](file:///C:\Users\User\Desktop\ITRW324%20FINAL%20PROJECT.docx#_Toc26365665)

[Figure 4: ENTER PERSONAL DETAILS 1](file:///C:\Users\User\Desktop\ITRW324%20FINAL%20PROJECT.docx#_Toc26365666)

[Figure 5: TABLE SHOWING ALL PERSONAL CONTACTS 1](file:///C:\Users\User\Desktop\ITRW324%20FINAL%20PROJECT.docx#_Toc26365667)

[Figure 6:MESSAGE THAT PROMPTS CONFIRMATION 1](file:///C:\Users\User\Desktop\ITRW324%20FINAL%20PROJECT.docx#_Toc26365668)

[Figure 7: FORM FOR EDITING PERSONAL CONTACTS 1](file:///C:\Users\User\Desktop\ITRW324%20FINAL%20PROJECT.docx#_Toc26365669)

[Figure 8: CONFIMATION MESSAGE OF CONTACT DELETED 1](file:///C:\Users\User\Desktop\ITRW324%20FINAL%20PROJECT.docx#_Toc26365670)

[Figure 9: BUSINESS CONTACT INFROMATION INSERT 1](file:///C:\Users\User\Desktop\ITRW324%20FINAL%20PROJECT.docx#_Toc26365671)

# Reference list:

Anon.(2016).*Node.js.Introduction.*.Available: https://www.w3schools.com/nodejs/nodejs\_intro.asp. Last accessed 1 Dec 2019.

Anon. (2016). *Vue.js.* Available: https://www.w3schools.com/whatis/whatis\_vue.asp. Last accessed 1 Dec 2019.