Programming Assignment #1 - SQL Contact List

by Habil Damania
Net-ID: hxd170005

1. Introduction

With the growth of digitization and increased social interaction amongst people, it has become cumbersome to remember the details of our close relatives, friends, employee's, etc. The old way of writing the address details of a person in a diary has also become obsolete. A new digital way of storing the contact list of people has come into prominence which helps provide integrity, confidentiality and availability of data at all times and saves the hassle of searching through a telephone diary to find the details of a person. It also avoids the trouble of keeping the diary in a safe place since everything you need is available through your mobile or computing device.

In this programming project, a web based contact list application is implemented to store the contact details of a person including but not limited to the persons Full Name, Address, Phone Number, Birth Date, etc. This application can be hosted on an online server which would give the user the freedom to access the contact list from anywhere and on any device. The application provides various widgets to ensure smooth and user-friendly operation of the contact list and enforces usability and ease of operation at the same time.

What differs this application from traditional contact list apps is the flexibility to search for any combination of names, phone numbers, addresses, dates, etc using just a single search bar which dynamically displays the result in real time (Just Like Google's Search engine). The User is also provided with the flexibility to Modify or Delete an entry dynamically from the retrieved search results all in real time. This increases the usability and saves on a lot of the user's time. The user is also provided the flexibility to Add an Entry into the Contact list by clicking the Add button on the top of the page. By doing so, a form in the form of a dialog is popped up where the user can enter the contact details and add an entry. On adding an entry into the contact list, results are reflected in real time without loading or redirecting on a new page.

The beauty of the application is that it is constrained to just one web page and there are no redirections to any new pages. In other words, the operations on the application take place dynamically on the same page without the page reloading at any time (saving a lot of time). The contact list is displayed in a tabular format which is mobile and computer responsive and provides the user the ability to scroll down the list or simply search using the search bar. A widget button is provided at the bottom of the screen which when clicked will take the user back to the top of the page avoiding the user to manually scroll all the way up.

Therefore, with the aforementioned features and many more intricate features, this application serves to be a perfect replacement for the now obsolete Contact books. The following sections describe the application in more detail

The completed Software project has met all the following required criteria to the best of my knowledge:

- Your host application should have a GUI interface. You may implement either a native GUI application or a web GUI interface.

- Your application GUI may be programmed with Java, Ruby, Python, Javascript, or PHP.
- Approved SQL databases are MySQL, PostgreSQL, MS SQL Server, and SQLite
- All interface with the database (queries, updates, deletes, etc.) must be done from a graphical user interface of your original design. Your GUI application will interface with the Contact List SQL database via an appropriate SQL connector. Initial database creation and population may be done from command line or any other admin tool, like a Workbench.
- Required information to display in a Contact Display Window includes: • First, Middle, and Last name • Address List – must accommodate a variable number of addresses, each of which should contain the following: - Type of address (e.g. home, work, other, etc.) - Street Address - City - State - Zip code (i.e. postal code) • Phone Number List - must accommodate a variable number of phone number, each of which should contain the following: - Type of number (e.g. home, work, fax, etc.) - Area Code (3-digits) - Number • Date List – must accommodate a variable number of dates, each of which should contain the following: - Type of date (e.g. birthday, anniversary, etc.) - Calendar date • An option to modify or delete an entry from the Contact Display Window.
- Using your GUI, be able to search for a contact, given any combination of Name components, Address component(s), Phone number components. All searches should be done via a single search field (like Google or Bing). The result of your query should display a list of all contacts that match the search.
- An option to modify or delete an entry from the search results.
- Your GUI should provide a menu item or button that brings up a New Contact Entry form. The New Contact Entry form should allow for entry of all data, including the ability to add a new Address, new Phone number, or new Date to the respective lists.
- Your GUI should provide a menu item or button that brings up a Modify Contact Entry form. The New Contact Entry form should allow for entry of all data, including the ability to add a new Address, new Phone number, or new Date to the respective lists.

1.1. System Overview:

The System implements a three-tiered architecture. The top tier also called as the user level layer is the GUI which is implemented using HTML, CSS5, and Bootstrap. The rendering of data in real time is done by using JQuery and AJAX. JQuery is used to cause an event on any user action in dynamic time and then use the Jquery.ajax() function to pass the data to other files for query processing and then retrieve the results using the same function. This provided the redirections and ensured dynamic processing of the web page.

The Second tier or middleware used is PHP. PHP is used to establish a connection to the MySQL database and then using the connection object to process the queries and perform operations on the database. It performs as the middleware to bind the GUI with the database at the backend. Query processing for operations such as updating, deleting, adding and searching are implemented in PHP for the user data obtained through the GUI using AJAX.

The final tier is the backend of the application which serves as the storage space for all the user contact data. The backend used was the MySQL server. PhpMyAdmin was the workbench used for database implementation.

2. Design Considerations

The following considerations were made during the designing and implementation process.

2.1. Assumptions and Dependencies

The following were the assumptions made before and after the design process:

- The Contacts.csv file provided by the instructor was not normalized and required various levels of normalization.
- Many NULL values existed in the Contacts.csv file and they were meant to be NULL. These values either did not exist for the particular tuple or were misplaced.
- The Contacts.csv was mapped to the database in various relations conforming to the provided schema by the instructor.
- The Contacts.csv file was mapped to the database through phpMyAdmin. Segregation of the area code from the phone numbers was done on the excel level using the command "+TRUE(A1, FIND("-",A1)) "
- None of the values of Contact_id should be NULL.
- The user system is compatible with HTML, CSS5 and Bootstrap and is running on Chrome, FireFox, Microsoft Edge, Safari and any other compatible browser.
- The system on which the application is hosted must have sufficient RAM and processing capabilities since the data contains thousands of entries.
- User should have basic knowledge of computer and/or mobile systems.
- Host system must have dependencies such as Apache, mySQL, installed and a working internet connection.

2.2. General Constraints

Any global limitations or constraints that have a significant impact on the design of the system's software (and describe the associated impact). Such constraints may be imposed by any of the following (the list is not exhaustive):

- Hardware or software environment
- End-user environment
- Availability or volatility of resources
- Standards compliance
- Interoperability requirements
- Interface/protocol requirements
- Data repository and distribution requirements
- Security requirements (or other such regulations)
- Memory and other capacity limitations

- Performance requirements
- Network communications
- Verification and validation requirements (testing)
- Other means of addressing quality goals
- Other requirements described in the requirements specification

2.3. Goals and Guidelines

- Proper Functioning of the System with minimum Lag
- Error Free
- Bug Free
- Free of Loopholes or any Backdoor entries.
- Excellent Usability
- Overall Functioning and memory management
- Interactive User Interface.
- Flawlessly performing all necessary functions.

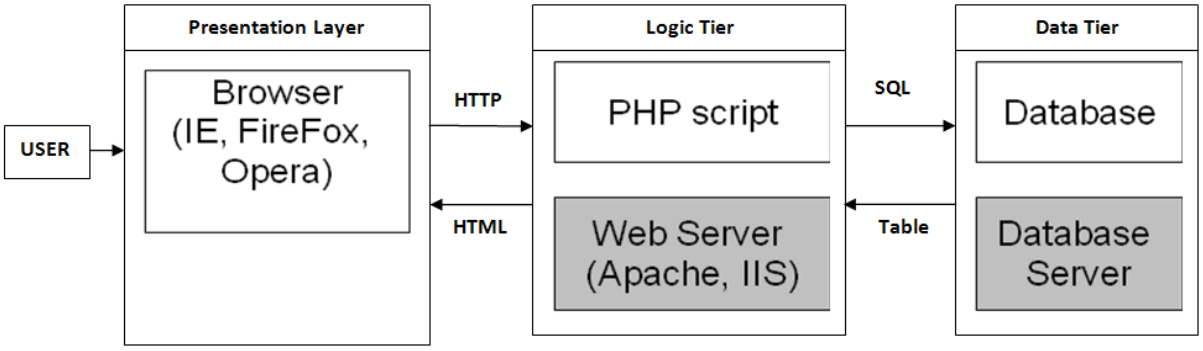
2.4. Development Methods

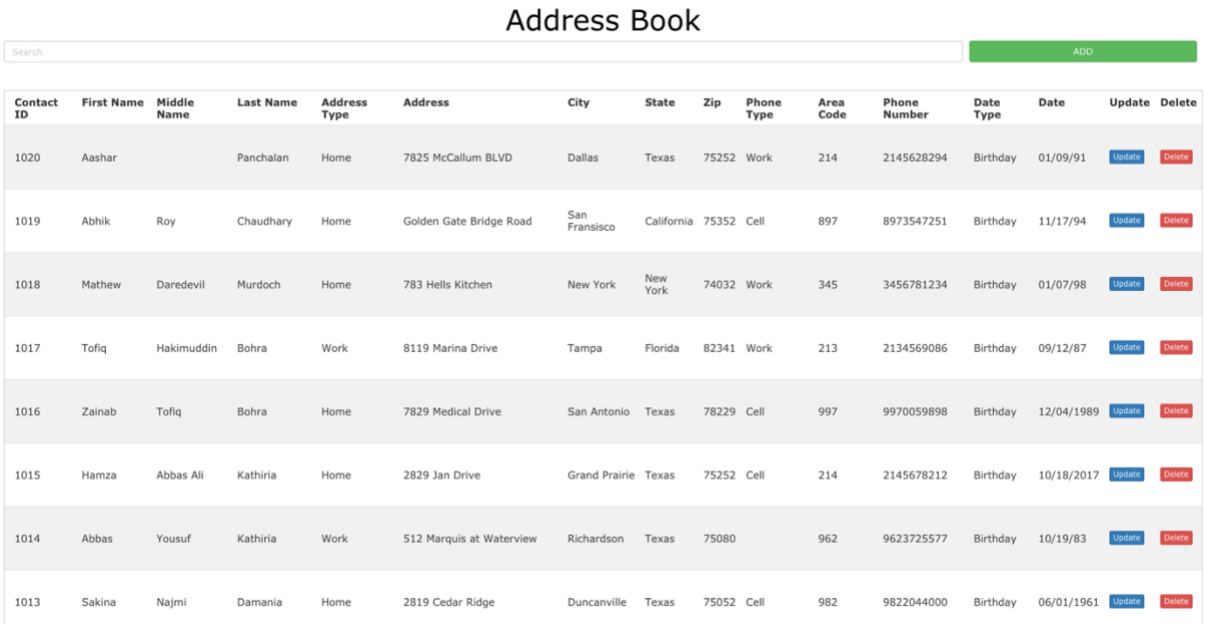
The Step wise development of the project is explained:

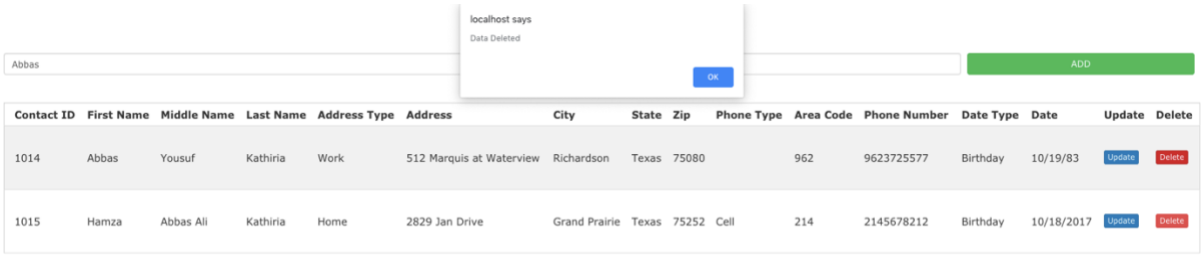
- First the Contacts.csv was populated into a database conforming to the schema provided by the instructor. The mapping of the csv file was done as follows: First the data in the csv file was segregated into multiple CSV files using excel commands as discussed above. Then each segregated CSV file was imported into the respective relation in the database using the import feature provided by phpMyAdmin.
- The Second Step involved creating a GUI for the system. This was done in HTML, CSS5. Various classes of tables, button types, modals, textbox were made use of in the GUI. W3schools.com proved to be a good source to find implementation strategies to build the GUI. W3 CSS classes were used to build responsive table in HTML.
- The third step involved data rendering. This step served the function of accepting user input on any event and sending the data to a PHP file for query processing. On successful execution of the PHP file, the output was pulled from the PHP file and rendered to the user in real time. This was implemented using the on.click() and jquery.ajax() functions in jquery. The implementation strategies and syntax of these methods were obtained from https://www.w3schools.com/jquery/ajax_ajax.asp
- The fourth step involved query processing. This was done using PHP. The data obtained from the user was sent to the PHP files using the POST method. The data was then pulled by the PHP side, and on that data query processing was done (Such as adding, deleting, modifying, or searching an entry). A connection was established to the database in PHP using the mysqli_connect(host, username, password, db_name) function.

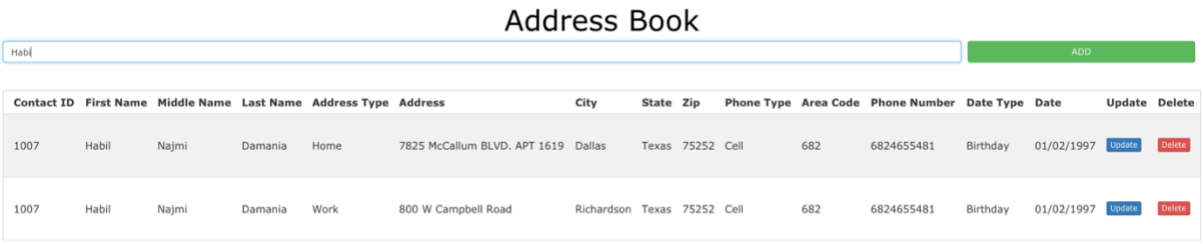
- Together all these modules were integrated to obtain a single integrated application serving the users purpose.
- Various types of Constraints are added at the user level. Some of them are, Phone number cannot be more than 10 digits, Area code cannot be more than 3 digits. A persons phone type can either be Home, Cell or Work. A persons address can either be Home or Work. Zip Code cannot exceed 5 numbers

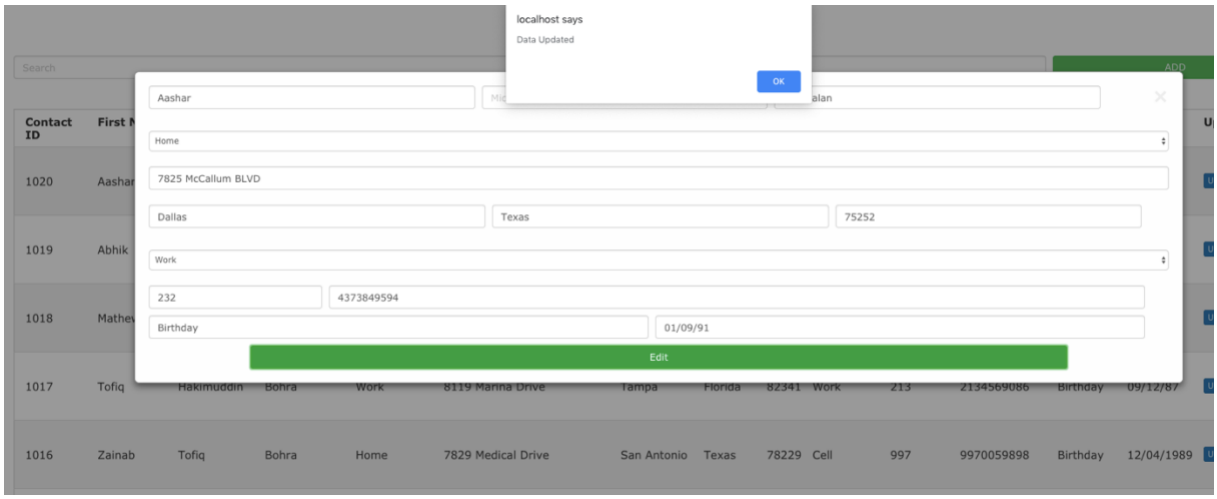
3. System Architecture











3.1. Resources

Following are the resources required:

- System with good RAM and processing power
- Apache, MySQL servers installed.
- MySQL workbench
- Fast Internet Connection
- Database to use with the application.

4. References

The following websites and links provided are the sources which have been referenced directly or indirectly during the implementation of this project. The material provided by these sources is open to use and modify.

- https://www.w3schools.com/w3css/w3css_tables.asp
- https://www.w3schools.com/howto/howto_css_modals.asp
- https://www.w3schools.com/jquery/ajax_ajax.asp
- https://www.w3schools.com/howto/howto_js_scroll_to_top.asp
- https://www.w3schools.com/html/default.asp
- https://www.w3schools.com/php/php_mysql_connect.asp