|  |
| --- |
|  |
| Contact List |
| CRUD Sample Code |
|  |
|  |
|  |

|  |
| --- |
| This application is built with ReactJs front-end, using the ReactJS create-react-app on Node.js, with an Express back-end. |

# Overview

When the page is visited a list of contacts is displayed. When a contact is clicked on their information is displayed, similar to a phone’s contact list. The user has the ability to view, add, update and delete contacts.

# Languages, APIs and Software

The application was written using the following:

**Languages and libraries**

* HTML
* CSS
* ReactJS
* JavaScript
* JQuery
* Express

**API and formats**

* Fetch API
* JSON

**Software**

* Node.js
  + create-react-app
* XAMPP
  + Apache
  + MySQL

# Functionality

1. Upon the page loading, an HTTP GET request is made, using the Fetch API, to retrieve all the contacts in the database.
2. Upon receipt of the contacts, a list is generated and displayed on the left side of the screen along with two options: add new contact and delete contact.
3. When the add new contact option is clicked a form is displayed on the right side of the screen with five inputs (first name, last name, phone number, email and address) and an add contact button.
   1. When the user leaves the phone number input, if the phone contains 10 digits it is re-formatted into (xxx) xxx-xxxx format.
   2. When the add button is clicked the information is validated: all inputs filled, name only has letters, phone is ten digits and email is a valid email format.
      1. If there are errors, a message is printed to the screen with the corrections that need to be made.
      2. If there are no errors, the data is sent to the server through a HTTP POST request using the Fetch API where it is added to the database. The view on the right is reset and a success message is displayed that the contact has been added.
4. If the user clicks on a contact’s name, that contact’s information is displayed on the right side of the screen in disabled inputs. Underneath the inputs are a delete contact button and an edit contact button.
   1. When the delete button is clicked, a HTTP DELETE request is made to the server using the Fetch API, where the contact and their information is removed from the database. The view on the right is reset and a success message is displayed that the contact has been removed.
   2. If the edit button is clicked, the five inputs are re-activated and allow for changes to be made. The edit button is also changed to display a save button.
      1. When the user leaves the phone number input, if the phone contains 10 digits it is re-formatted into (xxx) xxx-xxxx format.
      2. When the add button is clicked the information is validated: all inputs filled, name only has letters, phone is ten digits and email is a valid email format.
         1. If there are errors, a message is printed to the screen with the corrections that need to be made.
         2. If there are no errors, a HTTP PUT request is made to the server using the Fetch API where the database is updated to the new contact’s information. The view on the right is reset and a success message is displayed that the contact has been added.
5. If the user clicks “Delete Contact”, next to the contact’s name, a HTTP DELETE request is made to the server using the Fetch API, where the contact and their information is removed from the database. The view on the right is reset and a success message is displayed that the contact has been removed.