CONTACT LIST APPLICATION

SQL Programming Project

Overview

The Contact List application is a SQL programming project which involves the creation of database host application that interfaces with the backend SQL database.

Preethi Kesavan

UTD-ID: 2021486910

NET-ID: pxk190001

1. Introduction

The project is to create a contact list application for all users. The application should be able to display all contacts, add new contacts, modify or delete existing contacts and search for contacts.

This design document presents the designs used in implementing the project.

1.1 Purpose

The purpose of this document is to have a detailed description of the designs of the Contact List Application. This document fulfils one of the requirements of the programming assignment.

1.2 Scope

This document gives a detailed description of the software architecture of the contact list application. It specifies the structure and design of some of the modules

2. Design considerations

2.1 Assumptions

The user of the contact list application is aware of the basic operations of a computer and web pages. The user also understands the standard terms used for operation.

2.2 Constraints

The application is implemented using HTML, JavaScript, CSS on the client side and Node JS, Express framework on the server side. MySQL is used for database operations.

2.3 System environment

The web based contact list application is designed to work on all operating systems. The application can be downloaded from e-learning. The application requires the hosting system to have Node JS npm engine and MySQL installed. It also further requires the user to install all the dependency module mentioned in the readme.txt file for proper functioning of the application.

2.4 Design Methodology

The application is designed with flexibility for further development and/or modification. The application follows a Model-View-Controller architectural pattern that separates the application into three main logical components: the model, the view and the controller.

3. Software Interface Design

3.1 User Interface Design

The application user interface consists of four web pages -

- Home
- Display Contact
- Add Contact
- Search Contact

All the web pages in the application has a top bar that contains the links to the above mentioned web pages.

- The Home page contains the description of the application.
- The Display Contact page displays all the contacts present in the application with a edit and delete option beside it.
- The Add Contact page displays the form consisting of text and select fields. This page also has five buttons - Add Address, Add Phone, Add Date, Create and Cancel.
- The Search Contact field has a text field and a button. It also has a container to display the search results.

3.2 Module Interface Design

Module design maintains MVC (Model – View - Controller) architecture. View is the UI. Through UI user inputs data which goes to Controller. Controller transfers data into Model. If data is incorrect Model shows error message. Otherwise it processes the request, prepares the result and sends it to the Controller. Finally, Controller transfers generated code into View. The user views the result.

4. View

All the files corresponding to view module are present in the views folder inside the application.

The view includes all the html files -

- Home.html
- addContact.html
- searchContacts.html
- showContacts.html

The CSS for the whole application is given in - styles.css

The application also uses Bootstrap for which the css files are downloaded and present in the stylesheets folder under the views.

The scripts require to communicate with the server is also present in the scripts folder under the views. The script files include -

- contactScript.js
- getContact.js

5. Routes

The routes that are accessible in the application are mentioned in the index.js file inside the routes folder. The routes that involve some unique functionality are :

- '/'
- '/getAllContacts'
- '/Home'
- '/addContact'
- '/showContact'
- '/searchContact'
- '/getEditContact'
- '/editContact'
- '/deleteContact'
- '/createContact'
- '/searchInDB'

The following routes are used to load the scripts, stylesheets and images :

- '/stylesheets/style.css'
- '/stylesheets/images/background.png'
- '/stylesheets/images/linkbackground.jpg'
- '/stylesheets/bootstrap.min.css'
- '/scripts/contactScript.js'
- '/scripts/getContacts.js'
- '/scripts/bootstrap.min.js'

6. Controllers - Model

The Controller file used in the server side is present under Controllers folder. The file is ContactController.js
The routes mentioned in the previous section with their corresponding functionalities comprise of model - controllers.

i. '/' and '/Home'

Fetch and loads the home page of the application.
Input : None
Output : Home.html

ii. '/addContact'

Fetch and loads the add contact page of the application.
Input : None
Output : addContact.html

iii. '/showContact'

Fetch and loads the display contact page of the application.
Input : None
Output : showContacts.html

iv. '/searchContact'

Fetch and load the search contact page of the application.
Input : None
Output : searchContacts.html

v. '/getAllContacts'

Fetch all the contacts along with its details and displays the information in the specified container in showContact.html page.
Input : None
Output : JSON response of all contact entry containing all the fields of the contact in the database.

vi. '/createContact'

Send the new contact information from the addContact.html page and inserts the new contact in the database.
Input : POST data from the addContact.html containing information of First name, Middle name, Last name, Address(es), Phone number(s) and Date(s).
Output : JSON response of success or failure of contact addition.

vii. '/getEditContact'

Send the contact id from showContacts.html or searchContacts.html page to query the details of a particular contact in the database.
Input : Contact Id of the contact that is being edited.
Output : JSON response of the contact details of the mentioned contact.

viii. '/editContact'

Send the modified contact information from the showContacts.html or searchContacts.html page to save the changes in the database.
Input : POST data containing information of First name, Middle name, Last name, Address(es), Phone number(s) and Date(s).
Output : JSON response of success or failure of contact modification.

ix. '/deleteContact'

Send the contact id from showContacts.html or searchContacts.html page to delete the particular contact in the database.
Input : Contact Id of the contact that is being deleted.
Output : JSON response of success or failure of contact deletion.

x. '/searchInDB'

Send the search string from the searchContacts.html page to search for a matching contact(s) in the database. The search function works given any combination of Name components, Address component(s), Phone number components.
Input : String array containing the search strings
Output : JSON response of all the contacts with their details that matches the search criteria.

7. Database

MySQL is used as the database server. The queries used to initially create the database and corresponding tables are mentioned in the AllQueries.txt file in the public folder of the application.
Contacts.csv contains the data that is initially populated in the application when the application is installed and launched.