

Administrate Project

Introduction

The task is creating a web application that models an address book where the data persists even after restarting any processes. The address book should allow the user to view the names and contact details of people in organizations and insert, edit and delete the people and their details who are in an organization.

Design

Since the request includes persistent data, this application needs a database. Users will view through a user-interface on the client side. I need to process data and save it and that means a server is also required. I have used SQLite as the database along with Bootstrap, jQuery, CSS, HTML and Ajax requests in the front-end. I have also used Node.js with Express at the server-side of the application.

In my address book application, I wrote the server side using an MVC Framework. MVC helps for code reusability and ease of modification. This is to ensure that if the project needs to be extended in the future, it can be easily modified. This also allows for system maintenance to be easier.

I decided to use SQL database since it is a relational database and because I would be able to store, retrieve and edit data within it. I decided to go with SQLite due to its high portability as it will be tested and used in other PCs. SQLite was also simple; I did not need a client-server engine or any security functionality either.

For this application, I built a single-page application to reuse HTML and CSS. I achieved creating new views with Ajax requests by dynamically rewriting the current page. However, this method of implementation does not allow for navigation using links, such as "AddressBook/Administrate/Sara", instead of finding it from the table.

In the front-end, I used the Bootstrap framework to make the website responsive. This makes sure the page renders well on a variety of devices including mobile. I also used jQuery for writing simple Ajax requests and for easily selecting and manipulating DOM elements. I used a plug-in for the jQuery called DataTable which has pagination, search, column ordering and much more functionality for tables. Also, it was able to fill the table with an Ajax call.

Testing

I have tested many different test scenarios. I have fixed the bugs that I have encountered so far. These are some of the scenarios.

-Happy Path :

In the happy path scenario, I have entered contact details of new people, viewed and edited some of them.

-Edge Cases :

a) Using signs such as & < >

I have encountered a bug with in this case, since I was getting the "innerHTML" instead of "innerText" which I later realized it. I have fixed that.

b) Using Chrome source code enabling buttons and clicking them

This case was crashing the program. I have fixed that by implementing try and catch blocks for database queries to prevent wrong queries crashing the server.

c) Using spaces as Organization Name

My program was counting spaces as if they were organizations. I have fixed that by trimming the strings to make sure it will not get affected.

Conclusion

I believe I have created a comprehensive project in a short amount of time, but I think this project can be extended. If I had the access to the LinkedIn API (which needs a company page in LinkedIn). I could have used E-mail of contacts to pull their profile from LinkedIn and add them into the address book application. My project structure can easily extend for this additional functionality. Overall, I have implemented all the necessary requirements of the project successfully and tested for some cases.