L22.01-05 A Single Page Application (SPA) for our RESTful API

(L22.01)

Introduction

We will develop a full Single Page Application that consumes our already developed RESTful API using AJAX. We will implement the interface using a Javascript, HTML and CSS frameworks including, `jQuery` , `Bootstrap` , `Handlebars` and `FontAwesome` .

Setting up NodeJS modules and Frontend Frameworks

For this application we will require a "view engine" for NodeJS an ExpressJS. View Engines are server-side template engines which are used in conjunction with ExpressJS to populate HTML pages with information and data directly onto the view. This facilitates the generation of HTML dynamically (from templates). There are several popular engines, and I have used PUG already earlier in this module. Others include, EJS, Mustache and Handlebars. Handlebars is generally regarded as a super-set of Mustache as it can be used to do everything Mustache does, however, it also comes with additional logic. Handlebars also comes with compiled templates (compared to the interpreted templates of counterparts), consists of better path support when compared to Mustache, and has better support for global helpers than Mustache. We will Handlebars here as we only required a single view for this app. And it is a simple template.

For building, and styling the user interface (UI) HTML components we will use the Bootstrap Framework. And we will use jQuery as out JavaScript framework. We are familiar with the latter. Making interface elements such as Modals (including forms) is easy with Bootstrap.

I installed Handlebars `hbs` module for node using:

```
% cd quotations-app
% npm install --save hbs
```

An examining `package.json` shows

```
{
  "name": "quotations-app",
  "version": "1.0.0",
  "description": "Quotations App",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [
    "ExpressJS",
    "RESTful",
    "API",
    "MongoDB",
    "Quotations"
  ],
  "author": "John Keating",
  "license": "MIT",
  "dependencies": {
    "body-parser": "^1.19.0",
    "express": "^4.17.1",
    "hbs": "^4.1.1",
    "mongoose": "^5.9.10"
  }
}
```

You can see `hbs` in there so that's great. Next create the directories we require for the user interface (`public` and `views`), together with the `css` and `js` subdirectories:

```
% mkdir public
% mkdir views
% mkdir public/js
% mkdir public/css
```

I downloaded jQuery (https://jquery.com/download/), Bootstrap (https://getbootstrap.com/docs/4.4/getting-started/download/) and Font-Awesome (https://fontawesome.com/), and copied the minified and map files to the `js` and `css` subdirectories. Here is the updated directory structure:

```
% tree public
├── css
│   ├── all.css
│   ├── bootstrap-grid.min.css
│   ├── bootstrap-grid.min.css.map
│   ├── bootstrap.min.css
│   ├── bootstrap.min.css.map
│   └── fontawesome.min.css
├── js
│   ├── bootstrap.bundle.min.js
│   ├── bootstrap.bundle.min.js.map
│   ├── jquery-3.5.0.min.js
│   └── jquery-3.5.0.min.map
└── webfonts
    ├── fa-brands-400.eot
    ├── fa-brands-400.svg
    ├── fa-brands-400.ttf
    ├── fa-brands-400.woff
    ├── fa-brands-400.woff2
    ├── fa-regular-400.eot
    ├── fa-regular-400.svg
    ├── fa-regular-400.ttf
    ├── fa-regular-400.woff
    ├── fa-regular-400.woff2
    ├── fa-solid-900.eot
    ├── fa-solid-900.svg
    ├── fa-solid-900.ttf
    ├── fa-solid-900.woff
    └── fa-solid-900.woff2
```

Note that `/css/all.css` is the full Font-Awesome suite. I used this rather than the minified version. And notice I created a `/webfonts` directory to hold the font resources came with the Font-Awesome package. I used the `/js/bootstrap.bundle.min.js` Bootstrap resource as it includes `Popper.js` (https://popper.js.org/) bundled with Bootstrap for popups and dropdowns. Where possible I use the minified (`.min.js` or `.min.css`) versions of frameworks to reduce app load time.

Overview of Single Page Application App Interface (Functionality)

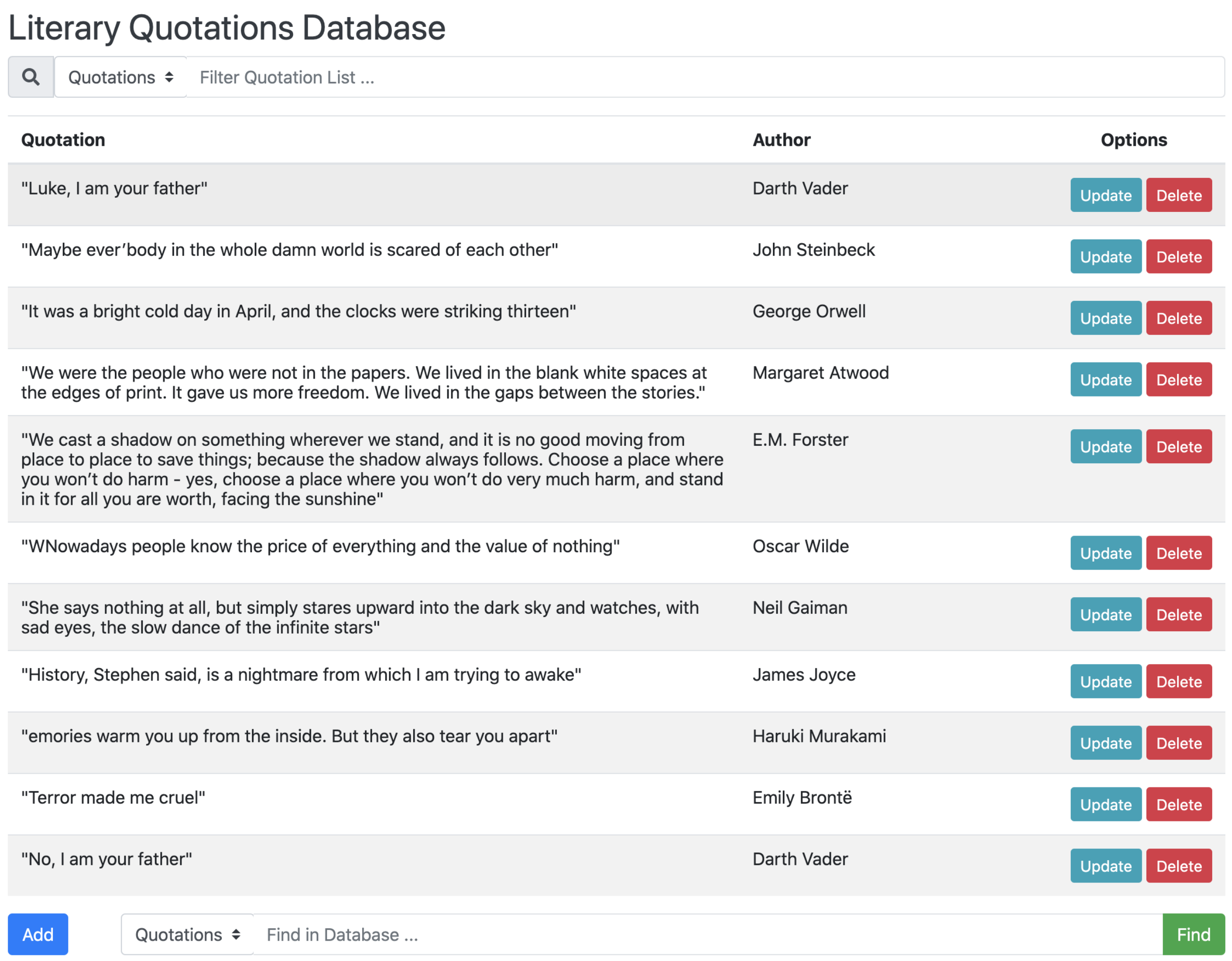
Here are some screenshots of the Single Page Application (SPA) user interface demonstrating the functionality. There is functionality within the app for managing the display and interaction with the results retrieved from the RESTful API. There is also functionality for adding new quotations, updating a quotation, and deleting a quotation. It is possible to search the database using a filter on the quote or author fields. And there is functionality provided to perform live searching (together with hiding non-matching table records) on results.

Working through the screenshots will give an overview of what we expect the view and the modals to look like (in advance of examining the code). The interface elements (supporting CRUD) have been Annotated using red boxes and are accompanied by textual descriptions (red font).

Almost all of the elements use the already created RESTful API, however, it was necessary to add a couple more routes and controllers for searching based on a search string. The app is independent of, and is said to consume the API. The app interface (structure and style)is independent of the of the API, and it is possible to build many different apps; you are limited by your imagination .. and HTML/JS/CSS coding skills, of course 😃 !

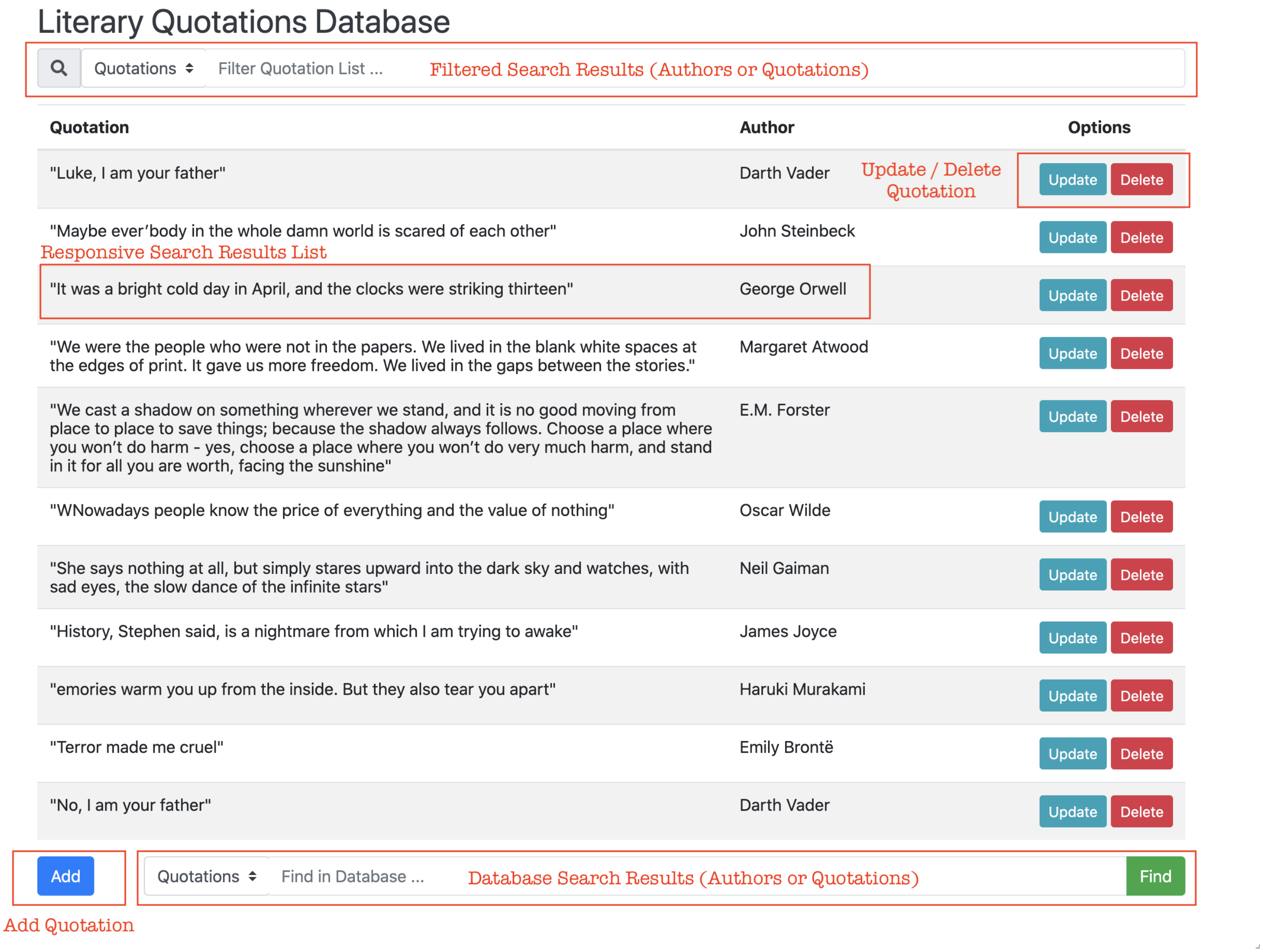
Let's look at the finished app:

Literary Quotations App



Quotation	Author	Options
"Luke, I am your father"	Darth Vader	Update Delete
"Maybe ever'body in the whole damn world is scared of each other"	John Steinbeck	Update Delete
"It was a bright cold day in April, and the clocks were striking thirteen"	George Orwell	Update Delete
"We were the people who were not in the papers. We lived in the blank white spaces at the edges of print. It gave us more freedom. We lived in the gaps between the stories."	Margaret Atwood	Update Delete
"We cast a shadow on something wherever we stand, and it is no good moving from place to place to save things; because the shadow always follows. Choose a place where you won't do harm - yes, choose a place where you won't do very much harm, and stand in it for all you are worth, facing the sunshine"	E.M. Forster	Update Delete
"WNowadays people know the price of everything and the value of nothing"	Oscar Wilde	Update Delete
"She says nothing at all, but simply stares upward into the dark sky and watches, with sad eyes, the slow dance of the infinite stars"	Neil Gaiman	Update Delete
"History, Stephen said, is a nightmare from which I am trying to awake"	James Joyce	Update Delete
"emories warm you up from the inside. But they also tear you apart"	Haruki Murakami	Update Delete
"Terror made me cruel"	Emily Brontë	Update Delete
"No, I am your father"	Darth Vader	Update Delete

Literary Quotations App (Annotated)



Literary Quotations Database

Filtered Search Results (Authors or Quotations)

Update / Delete Quotation

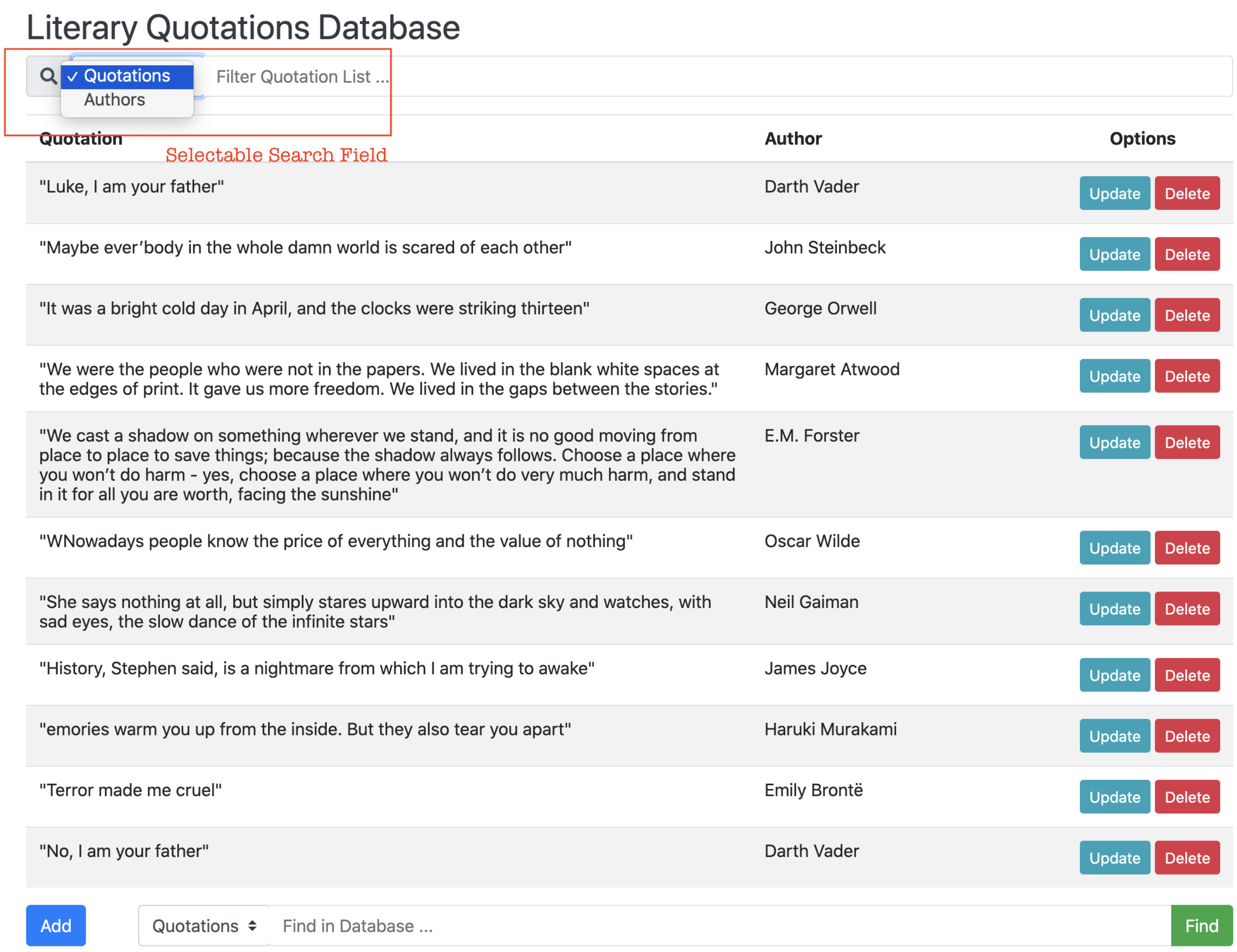
Responsive Search Results List

Database Search Results (Authors or Quotations)

Add Quotation

Quotation	Author	Options
"Luke, I am your father"	Darth Vader	Update Delete
"Maybe ever’body in the whole damn world is scared of each other"	John Steinbeck	Update Delete
"It was a bright cold day in April, and the clocks were striking thirteen"	George Orwell	Update Delete
"We were the people who were not in the papers. We lived in the blank white spaces at the edges of print. It gave us more freedom. We lived in the gaps between the stories."	Margaret Atwood	Update Delete
"We cast a shadow on something wherever we stand, and it is no good moving from place to place to save things; because the shadow always follows. Choose a place where you won’t do harm - yes, choose a place where you won’t do very much harm, and stand in it for all you are worth, facing the sunshine"	E.M. Forster	Update Delete
"WNowadays people know the price of everything and the value of nothing"	Oscar Wilde	Update Delete
"She says nothing at all, but simply stares upward into the dark sky and watches, with sad eyes, the slow dance of the infinite stars"	Neil Gaiman	Update Delete
"History, Stephen said, is a nightmare from which I am trying to awake"	James Joyce	Update Delete
"emories warm you up from the inside. But they also tear you apart"	Haruki Murakami	Update Delete
"Terror made me cruel"	Emily Brontë	Update Delete
"No, I am your father"	Darth Vader	Update Delete

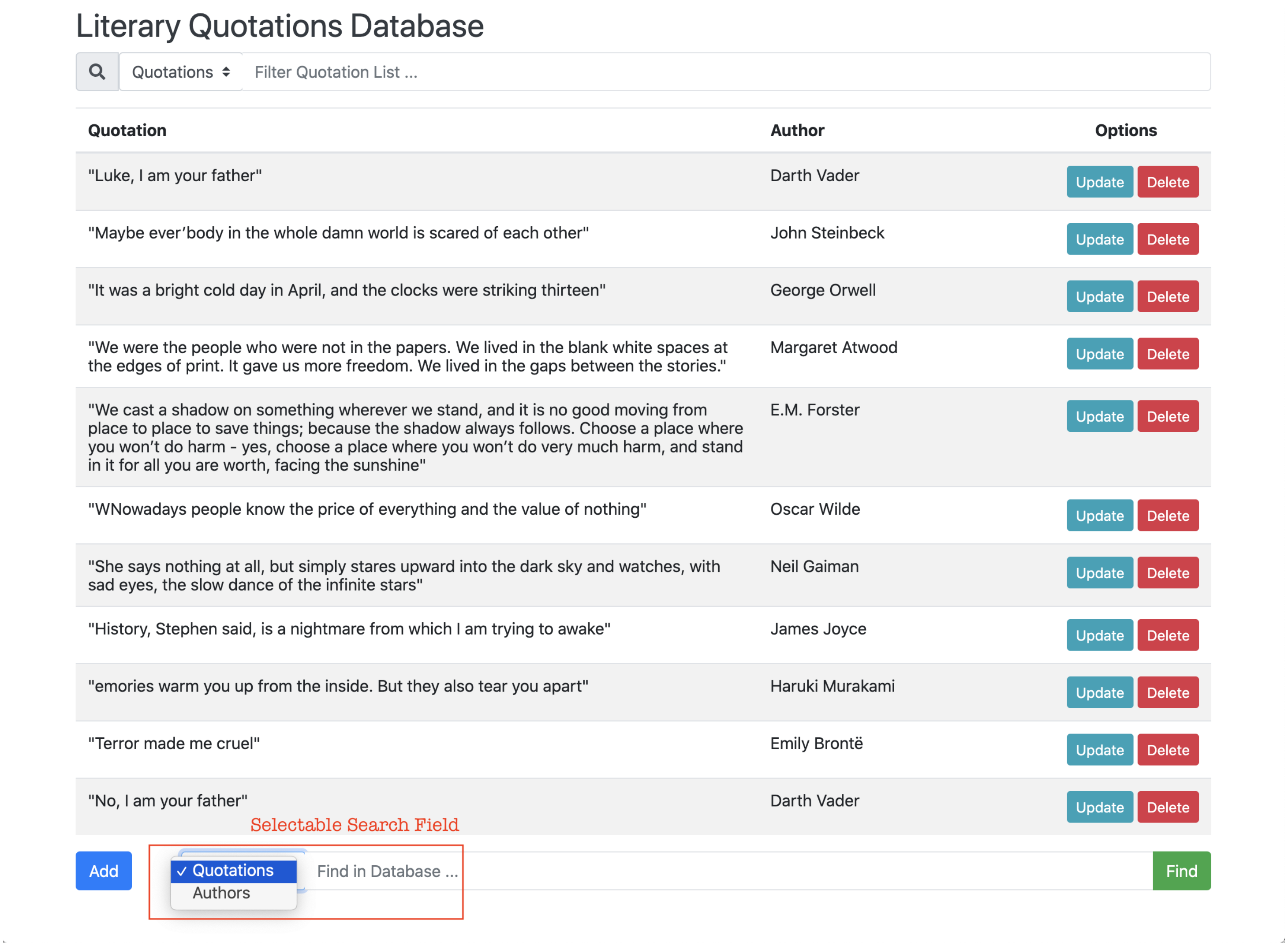
Results Filter (Quotations or Authors)



Literary Quotations Database

Quotation	Author	Options
"Luke, I am your father"	Darth Vader	Update Delete
"Maybe ever’body in the whole damn world is scared of each other"	John Steinbeck	Update Delete
"It was a bright cold day in April, and the clocks were striking thirteen"	George Orwell	Update Delete
"We were the people who were not in the papers. We lived in the blank white spaces at the edges of print. It gave us more freedom. We lived in the gaps between the stories."	Margaret Atwood	Update Delete
"We cast a shadow on something wherever we stand, and it is no good moving from place to place to save things; because the shadow always follows. Choose a place where you won’t do harm - yes, choose a place where you won’t do very much harm, and stand in it for all you are worth, facing the sunshine"	E.M. Forster	Update Delete
"WNowadays people know the price of everything and the value of nothing"	Oscar Wilde	Update Delete
"She says nothing at all, but simply stares upward into the dark sky and watches, with sad eyes, the slow dance of the infinite stars"	Neil Gaiman	Update Delete
"History, Stephen said, is a nightmare from which I am trying to awake"	James Joyce	Update Delete
"emories warm you up from the inside. But they also tear you apart"	Haruki Murakami	Update Delete
"Terror made me cruel"	Emily Brontë	Update Delete
"No, I am your father"	Darth Vader	Update Delete

Database Search Filter (Quotations or Authors)

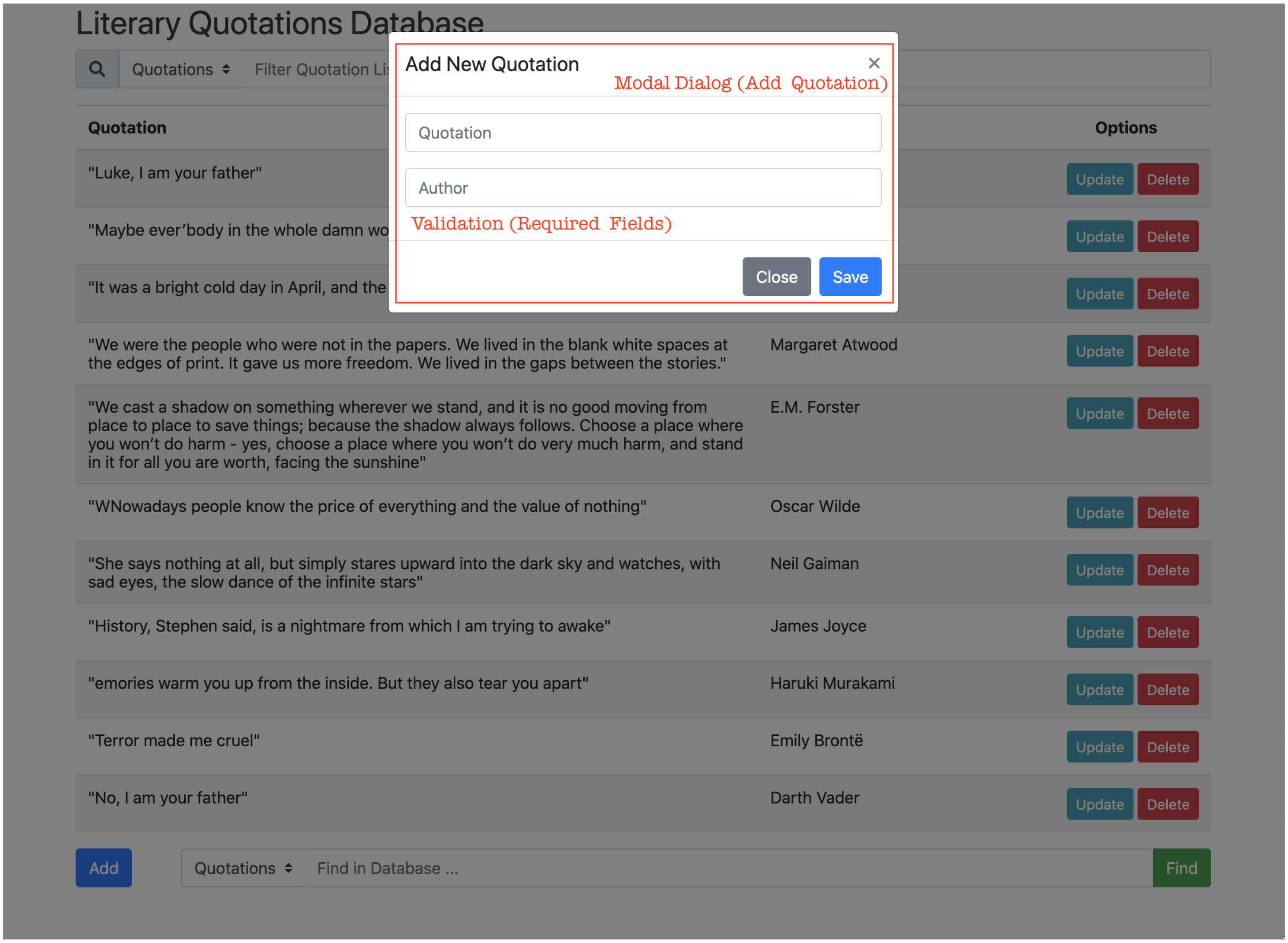


Literary Quotations Database

Quotation	Author	Options
"Luke, I am your father"	Darth Vader	Update Delete
"Maybe ever'body in the whole damn world is scared of each other"	John Steinbeck	Update Delete
"It was a bright cold day in April, and the clocks were striking thirteen"	George Orwell	Update Delete
"We were the people who were not in the papers. We lived in the blank white spaces at the edges of print. It gave us more freedom. We lived in the gaps between the stories."	Margaret Atwood	Update Delete
"We cast a shadow on something wherever we stand, and it is no good moving from place to place to save things; because the shadow always follows. Choose a place where you won't do harm - yes, choose a place where you won't do very much harm, and stand in it for all you are worth, facing the sunshine"	E.M. Forster	Update Delete
"WNowadays people know the price of everything and the value of nothing"	Oscar Wilde	Update Delete
"She says nothing at all, but simply stares upward into the dark sky and watches, with sad eyes, the slow dance of the infinite stars"	Neil Gaiman	Update Delete
"History, Stephen said, is a nightmare from which I am trying to awake"	James Joyce	Update Delete
"emories warm you up from the inside. But they also tear you apart"	Haruki Murakami	Update Delete
"Terror made me cruel"	Emily Brontë	Update Delete
"No, I am your father"	Darth Vader	Update Delete

Selectable Search Field

Add New Quotation

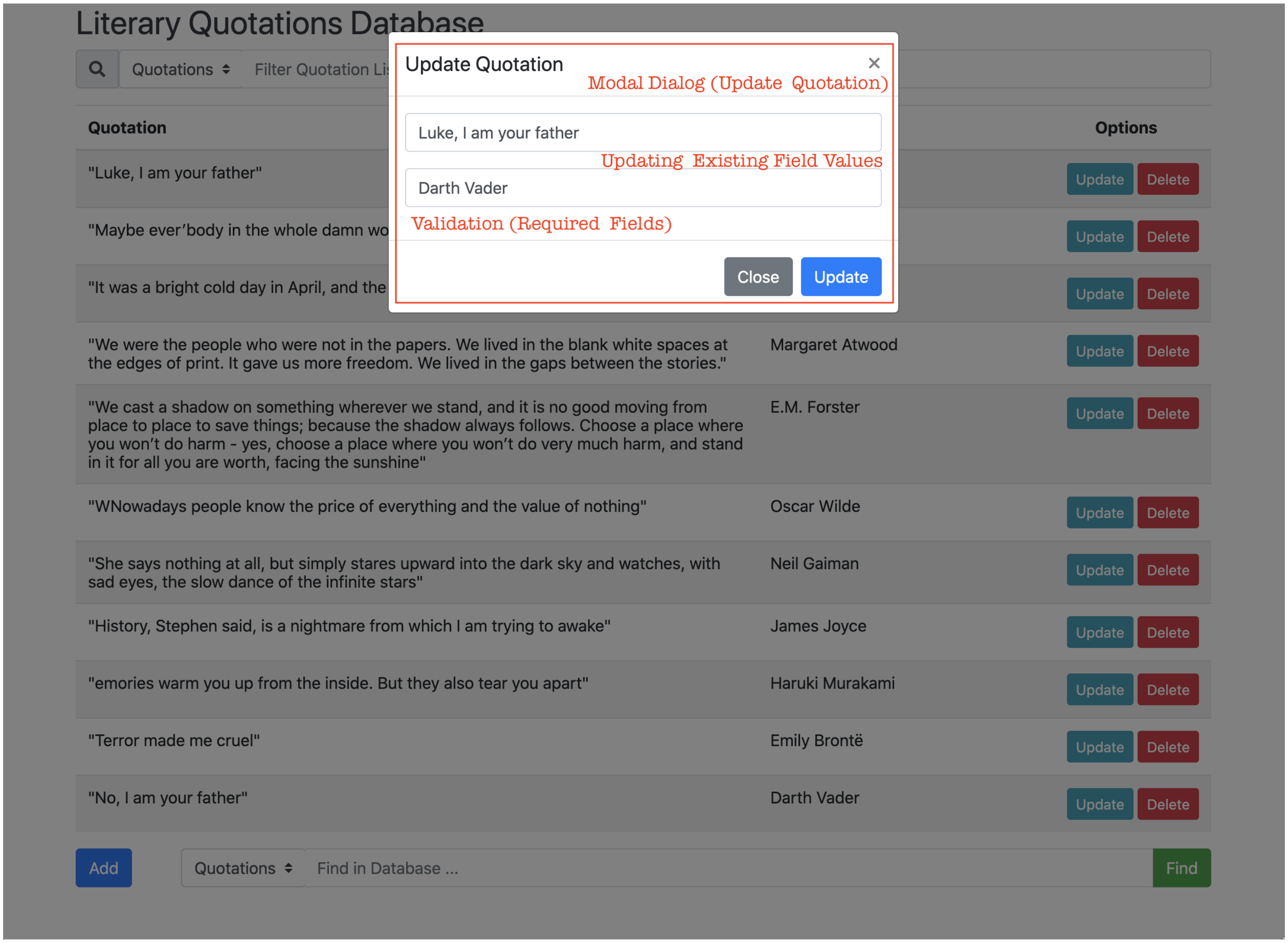


Modal Dialog (Add Quotation)

Validation (Required Fields)

Quotation		Options
"Luke, I am your father"		Update Delete
"Maybe ever'body in the whole damn wo		Update Delete
"It was a bright cold day in April, and the		Update Delete
"We were the people who were not in the papers. We lived in the blank white spaces at the edges of print. It gave us more freedom. We lived in the gaps between the stories."	Margaret Atwood	Update Delete
"We cast a shadow on something wherever we stand, and it is no good moving from place to place to save things; because the shadow always follows. Choose a place where you won't do harm - yes, choose a place where you won't do very much harm, and stand in it for all you are worth, facing the sunshine"	E.M. Forster	Update Delete
"WNowadays people know the price of everything and the value of nothing"	Oscar Wilde	Update Delete
"She says nothing at all, but simply stares upward into the dark sky and watches, with sad eyes, the slow dance of the infinite stars"	Neil Gaiman	Update Delete
"History, Stephen said, is a nightmare from which I am trying to awake"	James Joyce	Update Delete
"emories warm you up from the inside. But they also tear you apart"	Haruki Murakami	Update Delete
"Terror made me cruel"	Emily Brontë	Update Delete
"No, I am your father"	Darth Vader	Update Delete

Update Existing Quotation


Modal Dialog (Update Quotation)
Updating Existing Field Values
Validation (Required Fields)
×
Close
Update
Quotations ⬍
Filter Quotation Li
Add
Find

Literary Quotations Database

Update Quotation

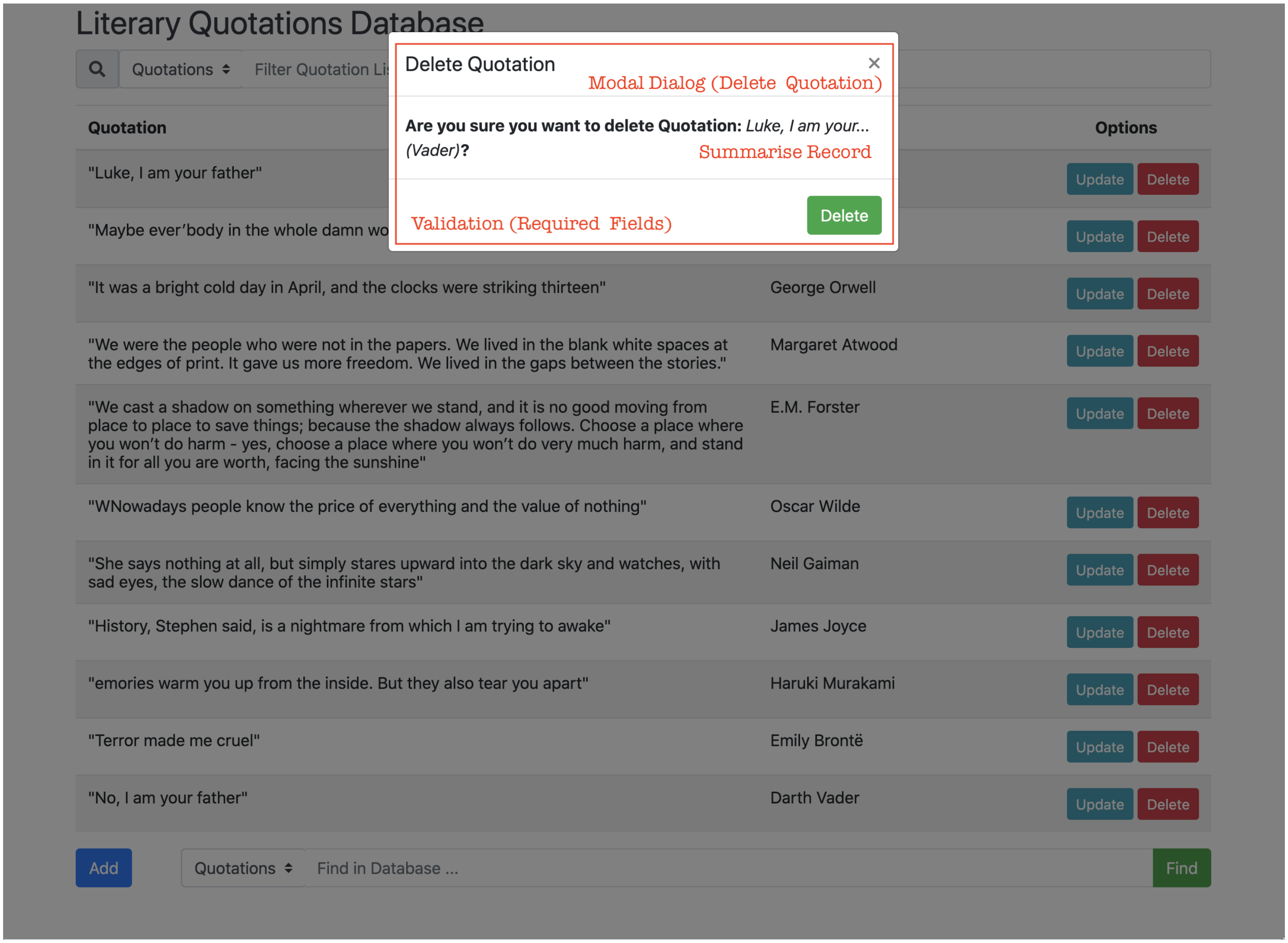
Luke, I am your father

Darth Vader

Quotation		Options
"Luke, I am your father"		Update Delete
"Maybe ever'body in the whole damn wo		Update Delete
"It was a bright cold day in April, and the		Update Delete
"We were the people who were not in the papers. We lived in the blank white spaces at the edges of print. It gave us more freedom. We lived in the gaps between the stories."	Margaret Atwood	Update Delete
"We cast a shadow on something wherever we stand, and it is no good moving from place to place to save things; because the shadow always follows. Choose a place where you won't do harm - yes, choose a place where you won't do very much harm, and stand in it for all you are worth, facing the sunshine"	E.M. Forster	Update Delete
"WNowadays people know the price of everything and the value of nothing"	Oscar Wilde	Update Delete
"She says nothing at all, but simply stares upward into the dark sky and watches, with sad eyes, the slow dance of the infinite stars"	Neil Gaiman	Update Delete
"History, Stephen said, is a nightmare from which I am trying to awake"	James Joyce	Update Delete
"emories warm you up from the inside. But they also tear you apart"	Haruki Murakami	Update Delete
"Terror made me cruel"	Emily Brontë	Update Delete
"No, I am your father"	Darth Vader	Update Delete

Quotations ⬍ Find in Database ...

Delete a Quotation



Literary Quotations Database

Delete Quotation ×

Modal Dialog (Delete Quotation)

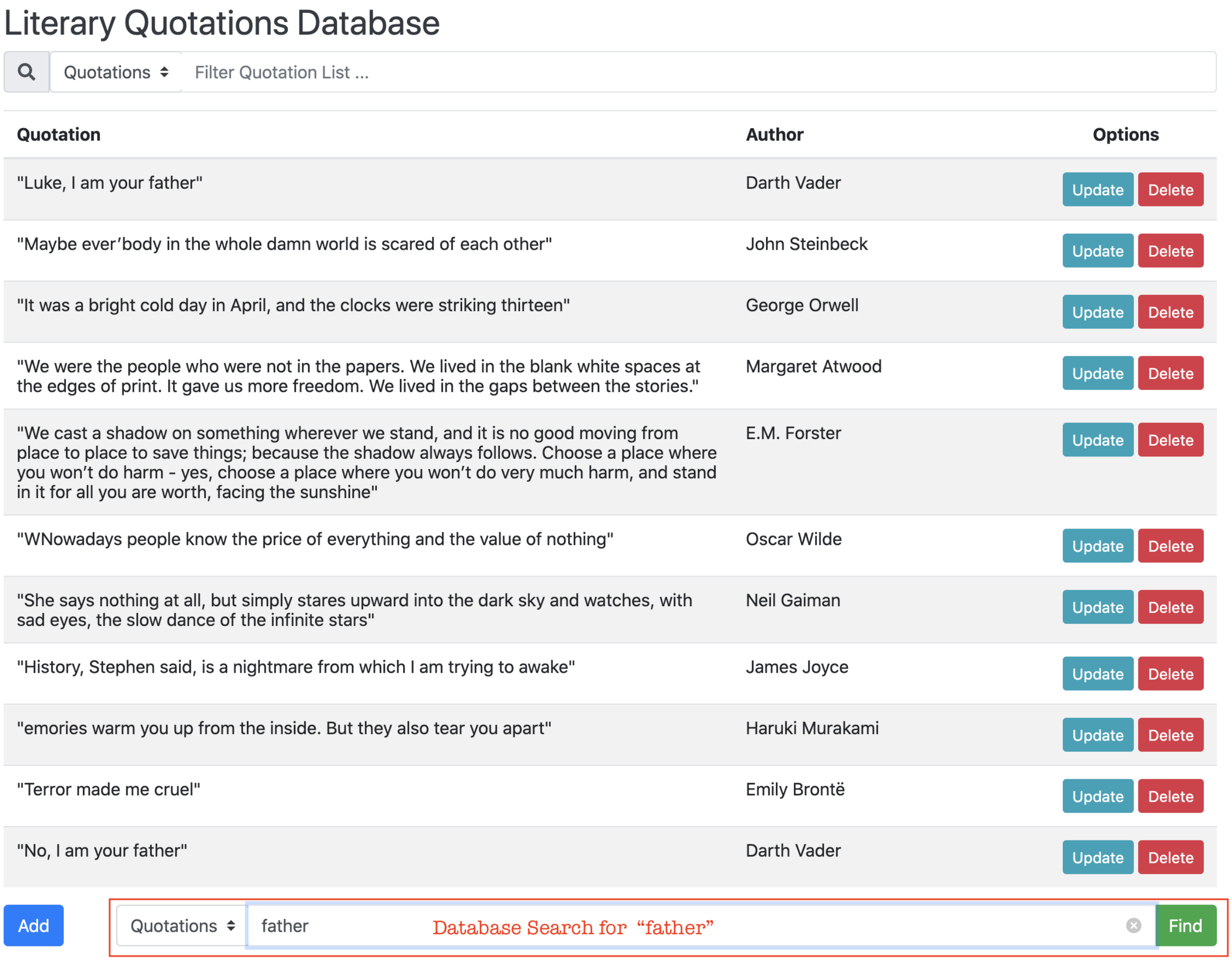
Are you sure you want to delete Quotation: *Luke, I am your... (Vader)***?**

Summarise Record

Validation (Required Fields)

Quotation		Options
"Luke, I am your father"		Update Delete
"Maybe ever'body in the whole damn wo		Update Delete
"It was a bright cold day in April, and the clocks were striking thirteen"	George Orwell	Update Delete
"We were the people who were not in the papers. We lived in the blank white spaces at the edges of print. It gave us more freedom. We lived in the gaps between the stories."	Margaret Atwood	Update Delete
"We cast a shadow on something wherever we stand, and it is no good moving from place to place to save things; because the shadow always follows. Choose a place where you won't do harm - yes, choose a place where you won't do very much harm, and stand in it for all you are worth, facing the sunshine"	E.M. Forster	Update Delete
"WNowadays people know the price of everything and the value of nothing"	Oscar Wilde	Update Delete
"She says nothing at all, but simply stares upward into the dark sky and watches, with sad eyes, the slow dance of the infinite stars"	Neil Gaiman	Update Delete
"History, Stephen said, is a nightmare from which I am trying to awake"	James Joyce	Update Delete
"emories warm you up from the inside. But they also tear you apart"	Haruki Murakami	Update Delete
"Terror made me cruel"	Emily Brontë	Update Delete
"No, I am your father"	Darth Vader	Update Delete

Search Database for Quotations matching Filter

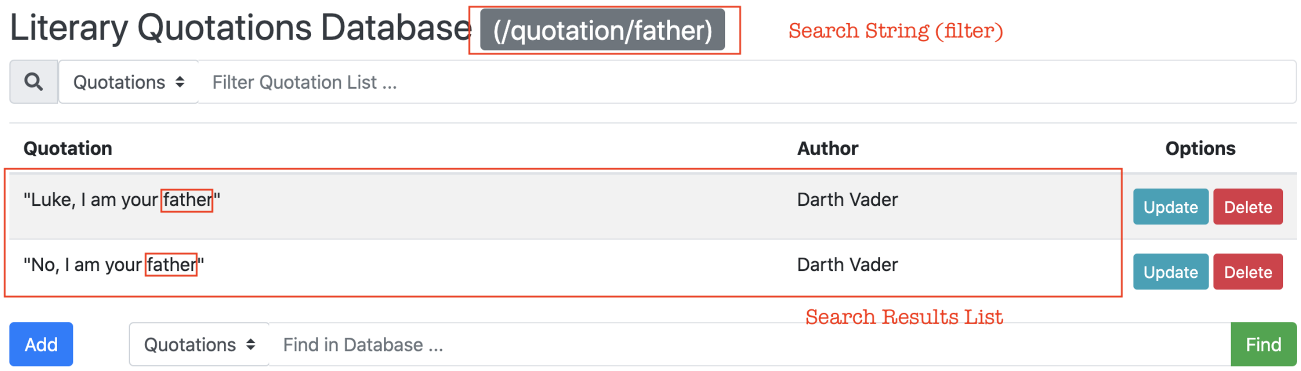


Literary Quotations Database

Quotation	Author	Options
"Luke, I am your father"	Darth Vader	Update Delete
"Maybe ever'body in the whole damn world is scared of each other"	John Steinbeck	Update Delete
"It was a bright cold day in April, and the clocks were striking thirteen"	George Orwell	Update Delete
"We were the people who were not in the papers. We lived in the blank white spaces at the edges of print. It gave us more freedom. We lived in the gaps between the stories."	Margaret Atwood	Update Delete
"We cast a shadow on something wherever we stand, and it is no good moving from place to place to save things; because the shadow always follows. Choose a place where you won't do harm - yes, choose a place where you won't do very much harm, and stand in it for all you are worth, facing the sunshine"	E.M. Forster	Update Delete
"WNowadays people know the price of everything and the value of nothing"	Oscar Wilde	Update Delete
"She says nothing at all, but simply stares upward into the dark sky and watches, with sad eyes, the slow dance of the infinite stars"	Neil Gaiman	Update Delete
"History, Stephen said, is a nightmare from which I am trying to awake"	James Joyce	Update Delete
"emories warm you up from the inside. But they also tear you apart"	Haruki Murakami	Update Delete
"Terror made me cruel"	Emily Brontë	Update Delete
"No, I am your father"	Darth Vader	Update Delete

Database Search for “father”

Filtered Database Search (for quotations including the word "father")



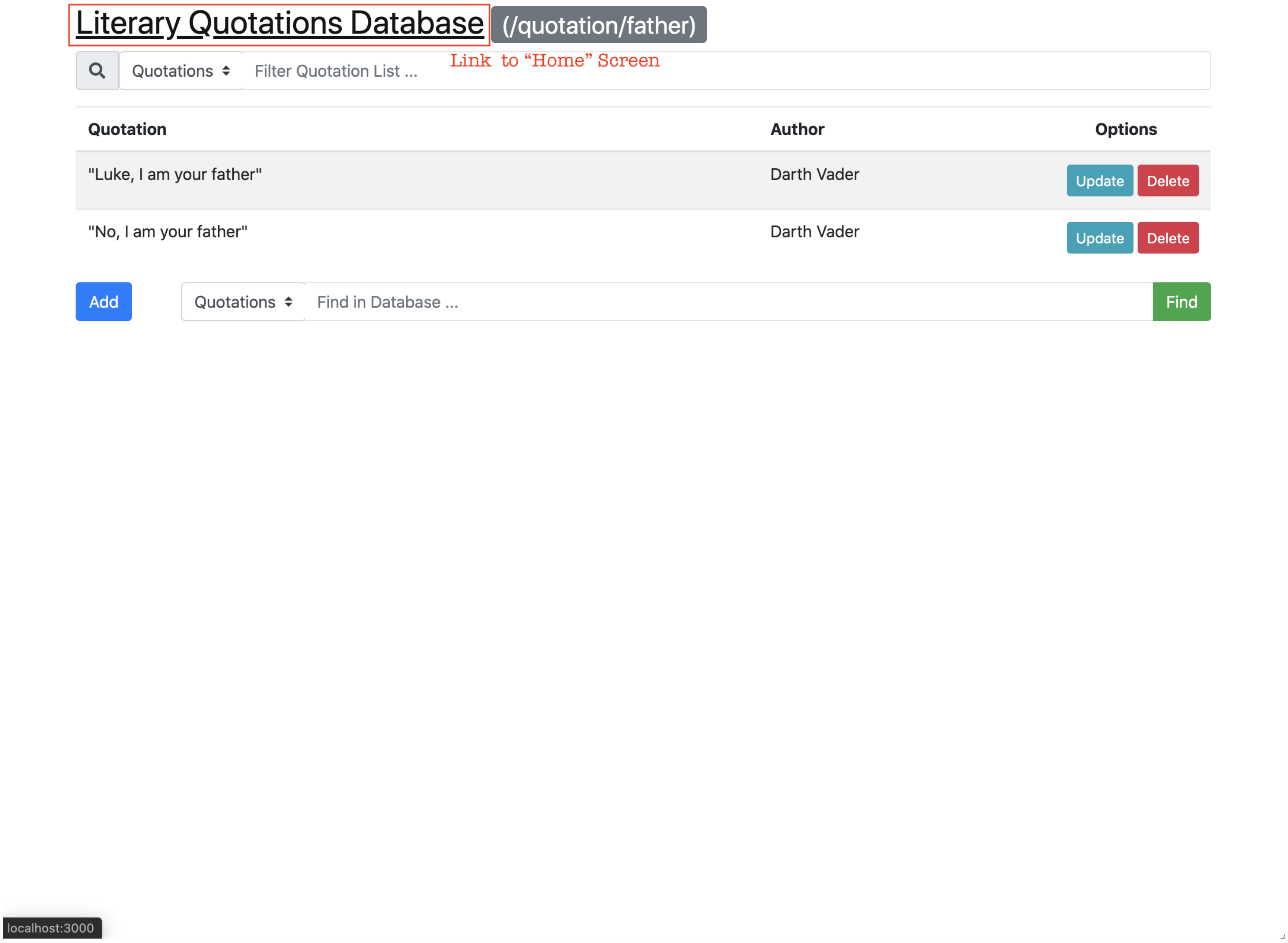
Literary Quotations Database (/quotation/father)

Search String (filter)

Quotation	Author	Options
"Luke, I am your father"	Darth Vader	Update Delete
"No, I am your father"	Darth Vader	Update Delete

Search Results List

Link to Home

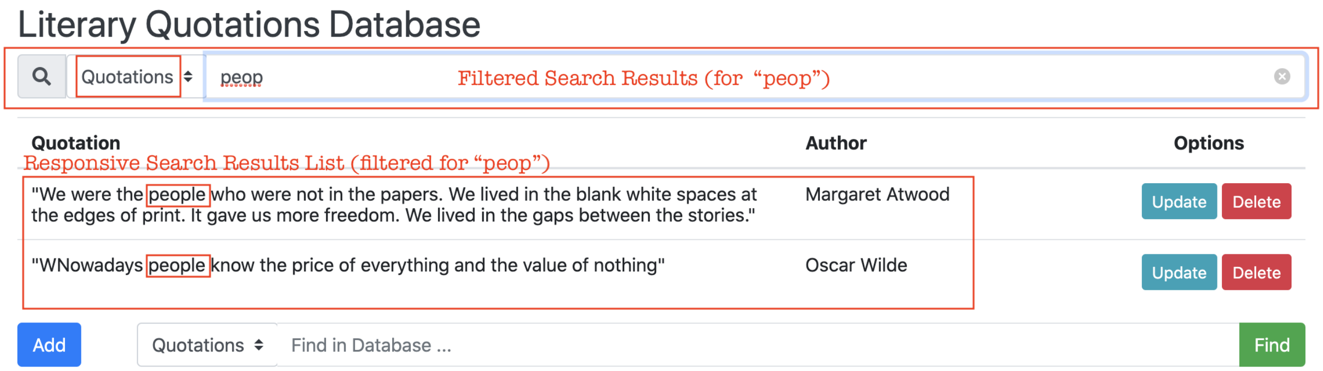


Literary Quotations Database (/quotation/father)

Link to “Home” Screen

Quotation	Author	Options
"Luke, I am your father"	Darth Vader	Update Delete
"No, I am your father"	Darth Vader	Update Delete

Filtered Results Search (for quotations including the string "peop")



Literary Quotations Database

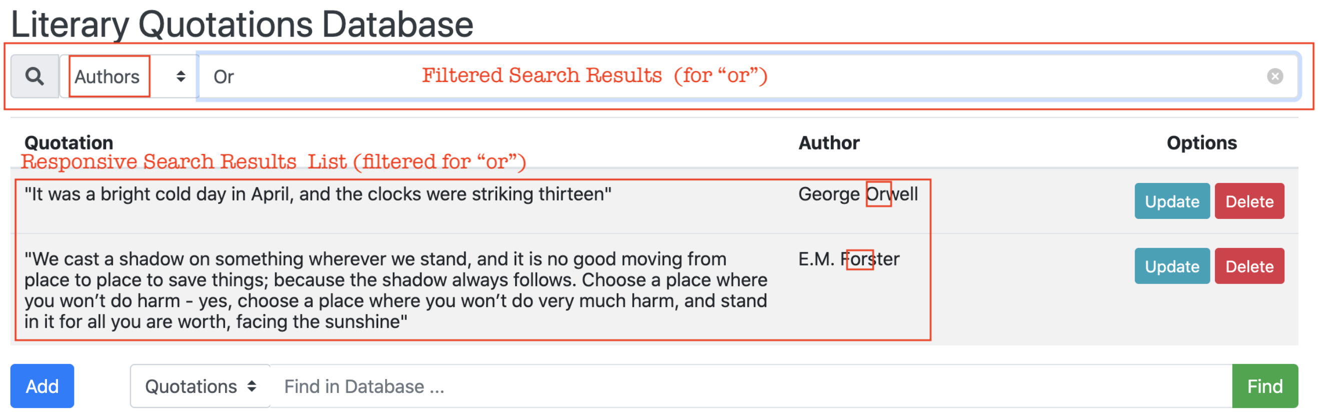
Quotations: peop

Filtered Search Results (for "peop")

Responsive Search Results List (filtered for "peop")

Quotation	Author	Options
"We were the people who were not in the papers. We lived in the blank white spaces at the edges of print. It gave us more freedom. We lived in the gaps between the stories."	Margaret Atwood	Update Delete
"WNowadays people know the price of everything and the value of nothing"	Oscar Wilde	Update Delete

Filtered Results Search (for authors including the string "or")



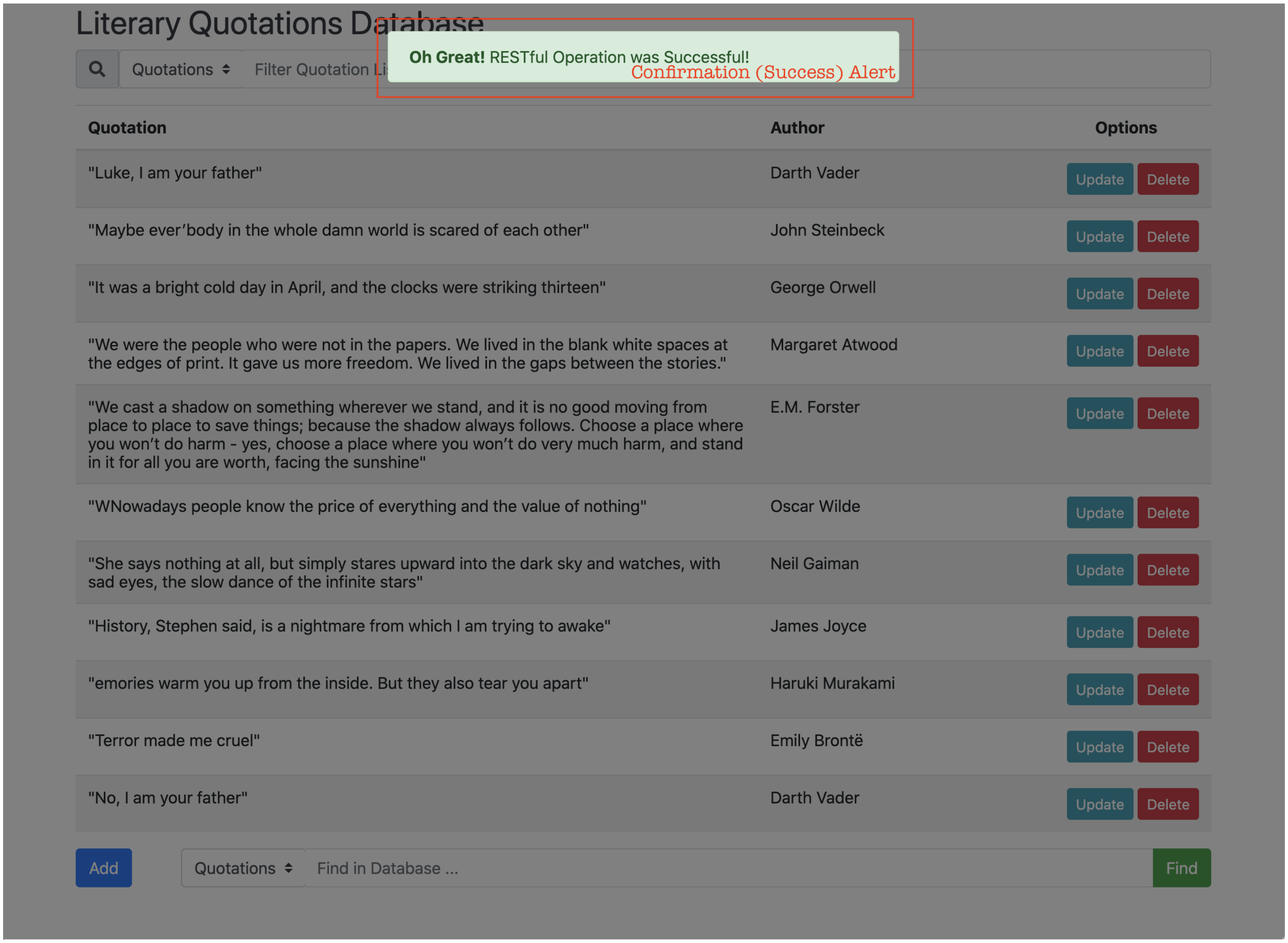
Literary Quotations Database

Filtered Search Results (for “or”)

Responsive Search Results List (filtered for “or”)

Quotation	Author	Options
"It was a bright cold day in April, and the clocks were striking thirteen"	George Orwell	Update Delete
"We cast a shadow on something wherever we stand, and it is no good moving from place to place to save things; because the shadow always follows. Choose a place where you won’t do harm - yes, choose a place where you won’t do very much harm, and stand in it for all you are worth, facing the sunshine"	E.M. Forster	Update Delete

Confirmation (Success) Alert



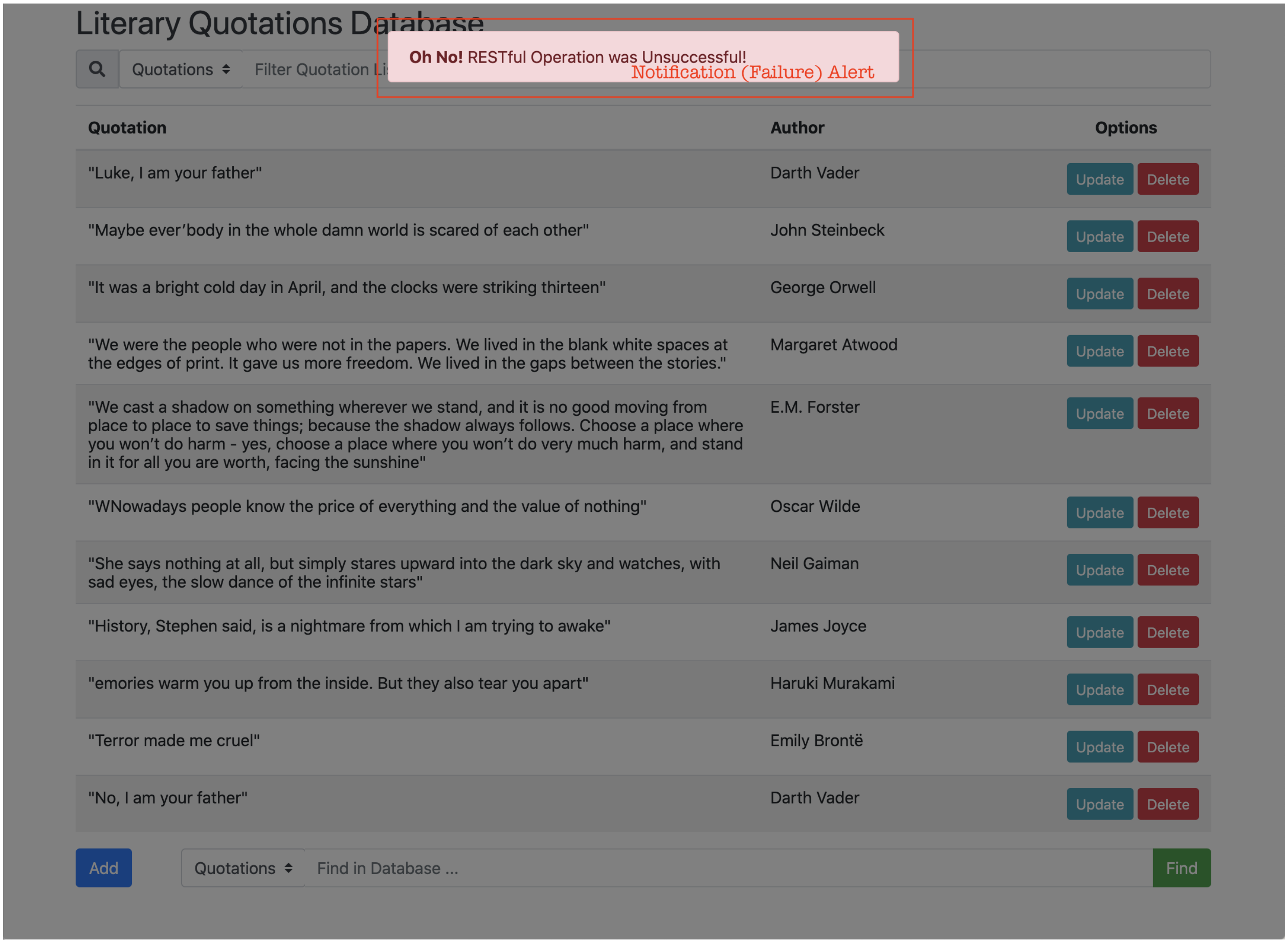
Literary Quotations Database

Oh Great! RESTful Operation was Successful!

Confirmation (Success) Alert

Quotation	Author	Options
"Luke, I am your father"	Darth Vader	Update Delete
"Maybe ever'body in the whole damn world is scared of each other"	John Steinbeck	Update Delete
"It was a bright cold day in April, and the clocks were striking thirteen"	George Orwell	Update Delete
"We were the people who were not in the papers. We lived in the blank white spaces at the edges of print. It gave us more freedom. We lived in the gaps between the stories."	Margaret Atwood	Update Delete
"We cast a shadow on something wherever we stand, and it is no good moving from place to place to save things; because the shadow always follows. Choose a place where you won't do harm - yes, choose a place where you won't do very much harm, and stand in it for all you are worth, facing the sunshine"	E.M. Forster	Update Delete
"WNowadays people know the price of everything and the value of nothing"	Oscar Wilde	Update Delete
"She says nothing at all, but simply stares upward into the dark sky and watches, with sad eyes, the slow dance of the infinite stars"	Neil Gaiman	Update Delete
"History, Stephen said, is a nightmare from which I am trying to awake"	James Joyce	Update Delete
"emories warm you up from the inside. But they also tear you apart"	Haruki Murakami	Update Delete
"Terror made me cruel"	Emily Brontë	Update Delete
"No, I am your father"	Darth Vader	Update Delete

Notification (Failure) Alert



Literary Quotations Database

Oh No! RESTful Operation was Unsuccessful!

Quotation	Author	Options
"Luke, I am your father"	Darth Vader	Update Delete
"Maybe ever'body in the whole damn world is scared of each other"	John Steinbeck	Update Delete
"It was a bright cold day in April, and the clocks were striking thirteen"	George Orwell	Update Delete
"We were the people who were not in the papers. We lived in the blank white spaces at the edges of print. It gave us more freedom. We lived in the gaps between the stories."	Margaret Atwood	Update Delete
"We cast a shadow on something wherever we stand, and it is no good moving from place to place to save things; because the shadow always follows. Choose a place where you won't do harm - yes, choose a place where you won't do very much harm, and stand in it for all you are worth, facing the sunshine"	E.M. Forster	Update Delete
"WNowadays people know the price of everything and the value of nothing"	Oscar Wilde	Update Delete
"She says nothing at all, but simply stares upward into the dark sky and watches, with sad eyes, the slow dance of the infinite stars"	Neil Gaiman	Update Delete
"History, Stephen said, is a nightmare from which I am trying to awake"	James Joyce	Update Delete
"emories warm you up from the inside. But they also tear you apart"	Haruki Murakami	Update Delete
"Terror made me cruel"	Emily Brontë	Update Delete
"No, I am your father"	Darth Vader	Update Delete

(L22.02)

Setting up View Handling using ExpressJS and Handlebars

Now that we have seen the view (interface) let's see how to implement it using ExpressJS, Bootstrap, Handlebars and jQuery.

To get started we need to create a `/views` subdirectory in our `quotations_app` root. We will create a single view file, called `quotations_view.hbs` , in this directory that will contain the HTML template. We will also map the `/assets` directory normally used with template engines onto our `/public` folder (already exists).

We will need to let ExpressJS know that it will be using Handlebars (`hbs`) as the view engine and we'll also need to set paths to directories. Here is the updated `server.js` file that accomplishes this. The updates to the RESTful API are wrapped in comments labeled as `/* == USER INTERFACE ADDITIONS == */`.

```
const express = require('express');         // we're making an ExpressJS App
const bodyParser = require('body-parser');  // we'll use body-parser extensively

const app = express();                      // create the ExpressJS app

/*  == USER INTERFACE ADDITIONS == */
const hbs = require('hbs');                 // use hbs view engine
const path = require('path');               // use the path module (for views)
/*  == USER INTERFACE ADDITIONS == */

// parse the different kinds of requests (content-type) the app handles
// use the "use" method to set up the body-parser middlewear
app.use(bodyParser.json())                          //  application/json
app.use(bodyParser.urlencoded({ extended: true }))  // pplication/x-www-form-urlencoded

/*  == USER INTERFACE ADDITIONS == */
app.set('views',path.join(__dirname,'views'));              // set the views directory
app.set('view engine', 'hbs');                              // set the view engine to hbs
app.use('/assets',express.static(__dirname + '/public'));   // set public folder as "static" for
/*  == USER INTERFACE ADDITIONS == */


// Set up Mongoose and our Database connection
const dbConnect = require('./config/connect.js');
const mongoose = require('mongoose');

// Set up connection to the database
mongoose.connect(dbConnect.database.url, {
    useNewUrlParser: true,
    useUnifiedTopology: true,
    useFindAndModify: false
}).then(() => {
    console.log("Successfully connected to the MongoDB database");
}).catch(err => {
    console.log('Unable to connect to the MongoDB database', err);
    process.exit();
});

require('./app/routes/quotation.routes.js')(app);

// listen for requests on port 3000
app.listen(3000, () => {
    console.log("Server listening on port 3000");
});
```

Apart from these **five** additional lines of code, there are no further changes to the earlier developed `server.js` file.

We previously included an ExpressJS route for our app root (`/`) which just returned all records (a JSON list). We will now use this route in conjunction with our view template to render the JSON as a HTML formatted table. We will use Handlebars rules to map the JSON records into HTML `<table>` rows and columns, and style the table using Bootstrap. Finally, we will use ExpressJS to render the template, following the call as it services the route `/` . We will add two additional controllers for routes that search quotations based on a quote (`/quotation/:s`) and author (`/author/:s`) search string (`:s`). We'll see the updated and new controllers later.

Here are the (small) changes to the App's routes file (`quotations.routes.js`); note that the updates to the wrapped in comments labeled as `/* == USER INTERFACE ADDITIONS == */` .):

```
module.exports = (app) => {
    const quotations = require('../controllers/quotation.controllers.js');

    /*  == USER INTERFACE ADDITIONS == */
    // not an addition per se but  we will now use this to
    // call an updated  controller to  render SPA view.
    app.get('/', quotations.root);
    /*  == USER INTERFACE ADDITIONS == */

    // Create a new Quotation
    app.post('/quotations', quotations.create);

    // Retrieve all Quotations
    app.get('/quotations', quotations.findAll);

    // Retrieve a single Quotation specified by quotationId
    app.get('/quotations/:quotationId', quotations.findOne);

    // Update a Quotation specified by quotationId
    app.put('/quotations/:quotationId', quotations.update);

    // Update a Quotation's quotation field specified by quotationId
    app.put('/quotations/quote/:quotationId', quotations.updateQuote);

    // Delete a Quotation specified by quotationId
    app.delete('/quotations/:quotationId', quotations.delete);

    /*  == USER INTERFACE ADDITIONS == */
    // Search for Quotations matching s
    app.get('/quotation/:s', quotations.searchQuotation);
    app.get('/author/:s', quotations.searchAuthor);
        /*  == USER INTERFACE ADDITIONS == */
}
```

And here are the controller updates, to file `quotation.controllers.js` . Note the `root` controller contains the code to render the view using the `quotations_view.hbs` template. The other updates just perform searches matching a search string provided by the user via the app interface. You can add the following anywhere in the `quotation.controllers.js` file.

```
/*  == USER INTERFACE ADDITIONS == */
// Default message for / (get)
exports.root = (req, res) => {
    Quotation.find()
    .then(quotations => {
        res.render('quotations_view',{
            results: quotations
          });
    }).catch(err => {
        res.status(500).send({
            message: err.message || "An error occurred while retrieving all Quotations."
        });
    });
};

// search for quotations, matching string on quote field
exports.searchQuotation = (req, res) => {
    var search = req.params.s;
    console.log("Searching Quotations: "+search)
    Quotation.find({ quotation: new RegExp(search,"ig")})
    .then(quotations => {
        res.render('quotations_view',{
            results: quotations
          });
    }).catch(err => {
        res.status(500).send({
            message: err.message || "An error occurred while retrieving all Quotations."
        });
    });
};

// search for quotations, matching string on author field
exports.searchAuthor = (req, res) => {
    var search = req.params.s;
    console.log("Searching Authors: "+search)
    Quotation.find({ author: new RegExp(search,"ig")})
    .then(quotations => {
        res.render('quotations_view',{
            results: quotations
          });
    }).catch(err => {
        res.status(500).send({
            message: err.message || "An error occurred while retrieving all Quotations."
        });
    });
};
/*  == USER INTERFACE ADDITIONS == */
```

When we use `render('quotations_view')` to indicate the view to be used, we also specify the data passed to the view handler; in this case we pass the `quotations` variable (JSON object) as `results` (i.e. handler will search for `results` object containing data to add to the template).

That's everything required to implement the template handling and additional routes required for our app. Note that the quotations searches are just GET requests providing a search string, for example, `/quotation/father` would return a JSON list of quotations matching the string `father` against the quote field, and `/author/orwell` would return a JSON list of quotations matching the string `orwell` against the author field. Note the search uses a regular expression, and ignores case. You could enhance the search functionality easily if you wish. And you can test these routes using Insomina or Postman as before.

(L22.03)

Rendering the View using Handlebars and Bootstrap

Now that we have everything set up to use Handlebars as our template engine, and can pass data from the RESTful API, we need to see how to make view template, and populate it with the data (`reaults`) from the API call to `/` .

Here is a simple view that just displays the data as a table. We will expand this later when we wish to include the modals, and JS functionality. Save it into a file called `quotations_view.hbs` , start your server and navigate to `http://localhost:3000/` . You should see the simple interface containing the table, and no interactive elements.

```
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>CRUD User Interface for NodeJS, ExpressJS and MongoDB REST API</title>
    <link href="/assets/css/bootstrap.min.css" rel="stylesheet" type="text/css"/>
    <link href="/assets/css/all.css" rel="stylesheet" type="text/css"/>
</head>
<body>
  <div class="container"> <!-- class="table-responsive text-nowrap (play with Bootstrap attribut
    <h2><a class="breadcrumb-item text-dark" href="/">Literary Quotations Database</a> <span cla

    <!-- Quotations Table -->
    <table class="table table-hover table-striped" id="quotationsList">
      <thead>
        <tr>  <!-- <tr class="d-flex"> -->
          <th style="width: 60%">Quotation</th>
          <th style="width: 25%">Author</th>
          <th class="text-center" style="width: 15%">Options</th>
        </tr>
      </thead>
      <tbody>
        {{#each results}}
        <tr >
          <td style="width: 60%">"{{quotation}}"</td>
          <td style="width: 25%">{{author}}</td>
          <td class="text-right" style="width: 15%">
            <a href="javascript:void(0);" class="btn btn-sm btn-info update" data-id="{{ id }}"
            <a href="javascript:void(0);" class="btn btn-sm btn-danger delete" data-id="{{ id }}
          </td>
        </tr>
        {{/each}}
      </tbody>
    </table>

   </div>
</body>
</html>
```

We see the templating functionality above, where iteration over the results object is accomplished using the `{{#each results}} ... {{/each}}` block, and access to `result` 's fields is via `{{ id }}` , `{{ quotation }}` and `{{ author }}` .

The Options buttons (for Updating and Deleting the individual record) are included here, but they are not functional. We will implement the interaction functionality in a later section when we look at attaching jQuery `onclick` handlers and Bootstrap modals. These have attached an `class` (`update` and `delete`) that can be used to locate all buttons for the purpose of adding the click functionality (rather than generating unique `id` s for every button). Notice the header also contains an empty badge - we will use this later to show the database search filters (as shown earlier):

```
<h2>
  <a class="breadcrumb-item text-dark" href="/">Literary Quotations Database</a>
  <span class="badge badge-secondary" id="databaseFilterNotice"></span>
</h2>
```

(L22.04)

Adding Search Functionality

Let's add live (table) search and database search functionality to this basic template. The live search will be placed at the top of the table, and the database search will be placed at the bottom, alongside the Add button.

Here is the (Bootstrap) HTML code for the live search; add it above the table in the view template:

```
<!-- Search the Results (Quotations) Table  -->
<div class="input-group mb-3">
    <div class="input-group-prepend">
        <label class="input-group-text" for="inputGroupSelect01"><i class="fa fa-search"></i></l
        <select class="custom-select" id="fieldSearchSelect">
            <option value="1" selected>Quotations</option>
            <option value="2">Authors</option>
        </select>
    </div>
    <input class="form-control py-2 border-left-0 border" type="search" placeholder="Filter Quot
</div>
```

Notice the `<i class="fa fa-search"></i>` . This is the Font-Awesome font icon for the magnifying glass. We are using Bootstrap grouping and styling for the buttons. Bootstrap uses `Popper.js` for the popups; we use the selection box to give the user the option to search on authors or quotes in the quotation. Both the input box and selection option have `id` s so that we can access them using jQuery. We will add the live search functionality using jQuery.

The code for the database search is similar and we add it after the table block:

```
<!-- Database Search -->
<div class="input-group mb-3">
    <!-- Add Quotation Button  -->
    <button class="btn btn-primary mr-5" data-toggle="modal" data-target="#addModal">Add</buttor
    <!-- Search the Database (Quotations) -->
    <div class="input-group-prepend">
         <select class="custom-select" id="databaseSearchSelect">
            <option value="quotation" selected>Quotations</option>
            <option value="author">Authors</option>
        </select>
    </div>
    <input class="form-control py-2 border-left-0 border" type="search" placeholder="Find in Dat
    <div class="input-group-append">
        <button class="btn btn-success" type="button" id="databaseSearch">Find</button>
    </div>
</div>
```

Here we include the `Add` button and give it an `id` (`#addModal`) so that we can locate it later to add click functionality (using jQuery) to open the add modal. Once again we have the option of searching quotations or authors. And the click functionality will be added afterwards using jQuery.

You can save this new view to test it but without the jQuery the functionality won't be present in the application. The jQuery (JavaScript) required for the live search and database search are shown below. We add the functionality to `$(document).ready(function(){ ... }` which will be called once all of the required HTML has been rendered. We also need to include the jQuery and Bootstrap framework JavaScript libraries before the application's JavaScript block.

You can add the following code to the view just after the database search block.

```
<script src="/assets/js/jquery-3.5.0.min.js"></script>
<script src="/assets/js/bootstrap.bundle.min.js"></script>
<script>
    $(document).ready(function(){

        //
        // Search: Field Selection
        //
        $("#fieldSearchSelect").on('change',function() {
            var k = $(this).val();
            if (k == 1) {
                $("#tableFilter").attr("placeholder", "Filter Quotation List ...");
            } else  {
                $("#tableFilter").attr("placeholder", "Filter Author List ...");
            }
            // $("#tableFilter").val("");
            $("#tableFilter").keyup();
        });

        //
        // Search: Table Filter
        //
        $("#tableFilter").keyup(function(){
            var filter = $(this).val().toUpperCase();
            $("#quotationsList").find('tr').each(function(rows) {
                if (rows !==  0) {
                    var row = $(this);
                    var searching = "";
                    if ($("#fieldSearchSelect").val() == 1) {
                        searching = row.find("td:first").text().toUpperCase();
                    }  else  {
                        searching = row.find("td:first + td").text().toUpperCase();
                    }
                    if (searching.indexOf(filter) != -1) {
                        row.show();
                    } else {
                        row.hide();
                    }
                }
            });
        });

        //
        // Search: Database search
        //
        $("#databaseSearch").on('click',function() {
            var search = $("#databaseSearchFilter").val().replace(/[|&;$%@"<>()+,^  ]/g, "");
            if (search) {
                $(location).attr('href', '/'+$("#databaseSearchSelect").val()+'/'+search);
            } else {
                $(location).attr('href', '/');
            }
        });
        //
        //  If there is a database filter then add a badge
        //
        if ($(location).attr('pathname') !== '/') {
           $('#databaseFilterNotice').text("("+$(location).attr('pathname')+")");
        }
    });
</script>
```

We first include an `onchange` event handler to change the placeholder text when we switch between searching for quotes and authors. And then simulate a (live search) `keyup` event to switch the search from one field to another on a change to the selected field. This is nice interaction design functionality and easy to implement. We convert both filter term and cell contents to uppercase for the matching. It is easy to add a checkbox, for example, to the form to allow control over case-sensitive matching.

For the database search, we use the fact that a normal browser request is `HTTP GET` by default so all we need to do is construct a URL containing the correct search route provided by the RESTful API, and redirect to that route. We perform some sanitisation on the search string first. Note this is not perfect, but we would also need to perform sanitisation on the API side anyway.

For the live search we add a `keyup` handler to the (`#tablefilter`) input box. This checks by iterating over all rows in the table to see if there is a match against the selected cell; the first cell (`"td:first"`) for quotes, and the second cell (`"td:first + td"`) for authors). If there is no match the table row is hidden, otherwise it is shown.

(L22.05)

Adding the Modals to complete User Interface

Let's add the modals (modal dialogs) to the application. We make extensive use of Bootstrap modals functionality here. We need three modals, and accompanying submission handlers: Add, Update, and Delete. We also need to set up the various button clicks, by adding `click` handlers using jQuery. We also need to set up the success and failure notifications

So there is quite a bit of code for this final part.

Here are the three core modals:

```
<!-- Modal Add Quotation-->
 <form id="addForm" action="/quotations" method="POST">
     <div class="modal fade" id="addModal" tabindex="-1" role="dialog" aria-labelledby="exampleM
      <div class="modal-dialog" role="document">
        <div class="modal-content">
          <div class="modal-header">
            <h5 class="modal-title" id="exampleModalLabel">Add New Quotation</h5>
            <button type="button" class="close" data-dismiss="modal" aria-label="Close">
              <span aria-hidden="true">&times;</span>
            </button>
          </div>
          <div class="modal-body">
            <div class="form-group">
                <input type="text" name="quotation" class="form-control" placeholder="Quotation"
            </div>
            <div class="form-group">
                <input type="text" name="author" class="form-control" placeholder="Author" requi
            </div>
          </div>
          <div class="modal-footer">
            <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
            <button type="submit" class="btn btn-primary">Save</button>
          </div>
        </div>
      </div>
     </div>
 </form>

 <!-- Modal Update Quotation -->
 <form id="updateForm" action="/quotations" method="PUT"> <!-- we can't PUT with HTML form - thi
    <div class="modal fade" id="updateModal" tabindex="-1" role="dialog" aria-labelledby="exampl
     <div class="modal-dialog" role="document">
       <div class="modal-content">
         <div class="modal-header">
           <h5 class="modal-title" id="exampleModalLabel">Update Quotation</h5>
           <button type="button" class="close" data-dismiss="modal" aria-label="Close">
             <span aria-hidden="true">&times;</span>
           </button>
         </div>
         <div class="modal-body">
           <div class="form-group">
               <input type="text" name="quotation" class="form-control quotation" placeholder="Q
           </div>
           <div class="form-group">
               <input type="text" name="author" class="form-control author" placeholder="Author"
           </div>
         </div>
         <div class="modal-footer">
           <input type="hidden" name="quotationId" class="id">
           <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
           <button type="submit" class="btn btn-primary">Update</button>
         </div>
       </div>
     </div>
    </div>
 </form>

 <!-- Modal Delete Quotation-->
 <form id="deleteForm" action="/quotations" method="DELETE"> <!-- we can't DELETE with HTML form
     <div class="modal fade" id="deleteModal" tabindex="-1" role="dialog" aria-labelledby="myMod
         <div class="modal-dialog">
            <div class="modal-content">
                <div class="modal-header">
                     <h5 class="modal-title" id="myModalLabel">Delete Quotation</h5>
                     <button type="button" class="close" data-dismiss="modal" aria-label="Close"
                </div>
                <div class="modal-body">
                     <strong>Are you sure you want to delete this Quotation?</strong>
                </div>
                <div class="modal-footer">
                     <input type="hidden" name="quotationId" class="form-control id" required>
                     <button type="submit" class="btn btn-success">Delete</button>
                </div>
             </div>
         </div>
     </div>
 </form>
```

The modals contain Bootstrap styled forms, and notice that they all include the following attributes on the form field: `action="/quotations" method="DELETE"`. As you probably know forms only handle `GET` and `POST` requests so what is happening here? The approach used here is to add an event handler for clicks on the submit buttons, and then immediately disable (prevent) the normal form handling operation, and replace it with an AJAX call that extracts these attributes to make the correct (AJAX) API call.

And we also use modals to show the notification, moving to either a success or failure modal depending on the result, directly after the modal submit (and API call, of course). We use the standard Bootstrap Alert notifications in these modals.

```
<!-- Operation Successful Alert Modal -->
<div class="modal fade" id="successAlert" tabindex="-1" role="dialog" aria-labelledby="exampleMo
    <div class="modal-dialog" role="document">
        <div class="alert alert-success" role="alert">
            <strong>Oh Great!</strong> RESTful Operation was Successful!
        </div>
    </div>
</div>

<!-- Operation Unsuccessful Alert Modal -->
<div class="modal fade" id="errorAlert" tabindex="-1" role="dialog" aria-labelledby="exampleModa
    <div class="modal-dialog" role="document">
        <div class="alert alert-danger" role="alert">
            <strong>Oh No!</strong> RESTful Operation was Unsuccessful!
        </div>
    </div>
</div>
</div>
```

Here are the jQuery handlers for the modals and RESTful API calls.

```
//
// Handle Add Modal form submit using AJAX PUT
//
var addForm = $("#addForm");
addForm.submit(function(e){
    e.preventDefault(); // prevent handling the normal form submit click!
    $.ajax({
        type: addForm.attr('method'),
        url:  addForm.attr('action'),
        data: addForm.serialize(), // get data from form!
        success: function (data) {
            console.log('Update successful!');
            // make a timed Bootstrap Alert on Success
            // then re-direct to "/"
            $('#successAlert').modal('show');
            var timer = setTimeout(function() {
                $(location).attr('href', '/')
            }, 3000);
        },
        error: function (data) {
            console.log('An error occurred.');
            // make a timed Bootstrap Alert on Error
            // then re-direct to "/"
            $('#errorAlert').modal('show');
            var timer = setTimeout(function() {
                $(location).attr('href', '/')
            }, 3000);
        },
    });
    $('#addModal').modal('hide');
});
//
// Set up the Update Modal - It is shown, and data passed, via button click
//
$('#quotationsList').on('click','.update',function(){
    var quotation_id = $(this).data('id');
    var quotation_quote = $(this).data('quote');
    var quotation_author = $(this).data('author');
    // alert (quotation_id+": "+ quotation_quote+" "+quotation_author);
    $('#updateModal').modal('show');
    $('.quotation').val(quotation_quote);
    $('.author').val(quotation_author);
    $('.id').val(quotation_id);
});
//
// Handle Update Modal form submit using AJAX PUT
//
var updateForm = $("#updateForm");
updateForm.submit(function(e){
    e.preventDefault(); // prevent handling the normal form submit click!
    var id = updateForm.find('input[name="quotationId"]').val();
    $.ajax({
        type: updateForm.attr('method'),
        url:  updateForm.attr('action')+"/"+id,
        data: updateForm.serialize(), // get data from form!
        success: function (data) {
            console.log('Update successful!');
            // make a timed Bootstrap Alert on Success
            // then re-direct to "/"
            $('#successAlert').modal('show');
            var timer = setTimeout(function() {
                $(location).attr('href', '/')
            }, 3000);
        },
        error: function (data) {
            console.log('An error occurred.');
            // make a timed Bootstrap Alert on Error
            // then re-direct to "/"
            $('#errorAlert').modal('show');
            var timer = setTimeout(function() {
                $(location).attr('href', '/')
            }, 3000);
        },
    });
    $('#updateModal').modal('hide');
});
//
// Set up the Delete Modal - It is shown, and data passed, via button click
//
$('#quotationsList').on('click','.delete',function(){
    var quotation_id = $(this).data('id');
    // let's remind the user which quotation they are deleting
    var quotation_quote = $(this).data('quote');
    var quotation_author = $(this).data('author');
    var subQuote = quotation_quote.substring(0,15);
    var authorSurname = quotation_author.split(" ").splice(-1)[0];
    $("#deleteModal .modal-body").html("<span class=\"font-weight-bold\">Are you sure you want t
    $('#deleteModal').modal('show');
    $('.id').val(quotation_id);
});
//
// Handle Delete Modal form submit using AJAX DELETE
//
var deleteForm = $("#deleteForm");
deleteForm.submit(function(e){
    // make everything look like a Single Page Application
    // consuming the RESTful API routes based on user selection
    e.preventDefault();  // prevent handling the normal form submit click!
    var id = deleteForm.find('input[name="quotationId"]').val();
    $.ajax({
        type: deleteForm.attr('method'),
        url:  deleteForm.attr('action')+"/"+id,
        success: function (data) {
            console.log('Deletion successful!');
            // make a timed Bootstrap Alert on Success
            // then re-direct to "/"
            $('#successAlert').modal('show');
            var timer = setTimeout(function() {
                $(location).attr('href', '/')
            }, 3000);
        },
        error: function (data) {
            console.log('An error occurred.');
            // make a timed Bootstrap Alert on Error
            // then re-direct to "/"
            $('#errorAlert').modal('show');
            var timer = setTimeout(function() {
                $(location).attr('href', '/')
            }, 3000);
        },
    });
    $('#deleteModal').modal('hide');
});
```

Note that we use `.show()` and `.hide()` on the modals, for example, `$('#updateModal').modal('show');` to show the Update modal. And when we click on a button in the table, say Update, we access the data (including `id`) in order to pass to the modal form elements to minimise typing, etc. All of the AJAX calls have success and error handling functionality. Notice also that for every form submission event (contained in `e`) we immediately prevent the normal functionality using `e.preventDefault()`.

You can add these to `$(document).ready(function(){ ... }` to add the functionality to our app.

Summary

We developed a Single Page Application that consumes our already developed RESTful API using AJAX.