# Design Document

ClevrBooks app

Accounting App

Based on Node.js and Vue.js

Contents

[Design Document 1](#_Toc84671573)

[Overview 3](#_Toc84671574)

[Development 5](#_Toc84671575)

[XAMPP Configuration 5](#_Toc84671576)

[Laravel Backend 7](#_Toc84671577)

[Database 9](#_Toc84671578)

[Create table migrations 10](#_Toc84671579)

# Overview

The idea behind this project is to eliminate the dependency from the Quicken application. It regularly requires to login to their server and then wants to install mobile application.

It also provides a great incentive to explore many new web development technologies.

The application will use ‘Node.js/Express’ for the backend, and Vue.js for the UI or frontend.

The ‘Node.js/Express’ framework is written in javascript and provides all the necessary tools to act as a backend for the ‘CRUD[[1]](#footnote-1)’ API and database access with sequelize ORM[[2]](#footnote-2).

This will allow to run the application as a standalone package and will be accessible through ‘localhost or IP address’ which doesn’t require virtual host setup and can be used with VPN connected.

The database will be either SQLite3 or hosted on the current Dublin mySQL server and the data will be populated by exporting all Quicken accounts to CVS files, and reimported to SQL from an Excel or Access VBA script.

CVS Files

Quicken

SQLite3

or

mySQL

Access Database

Excel Worksheet

The frontend will use the Vue.js framework, written in javascript.

The Vue.js frontend will be used to build a Single Page Application (SPA), that will provide authentication and access to the backend API.



# Development

The development IDE will be “Visual Studio Code” from Microsoft with the following extensions:

* ESLint
* Vetur
* Git History

All development will us SQLite3 as the main SQL database.

The project will be hosted on GitHub.

## Project Configuration

The first step consists in creating an empty project on the GitHub server. Once the empty project is created, on the Windows local machine, clone the project in a local directory. On Bangor, the folder is C:\Users\vaill\Documents\work\WebProjects\ClevrBooks

Once cloned, edit the README.md file, commit and push to GitHub.

The following sections will describe the development of the server backend based on node.js and express and the fronter user interface based on Vue.js (Vue3)

## NODE.js Backend (Server)

To build the backend server, first make sure that node.js is installed. The version used for this project is v14.18.0

The project will also use ‘nodemon’ to monitor code change and restart the server automatically.

>npm install -g nodemon

From the server subdirectory, initialize the project

>npm init

package name: (server)

    version: (1.0.0)

    description: Backend Server for ClevrBooks app

    entry point: (index.js) src/app.js

    test command:

    git repository:

    keywords:

    author: Claude Vaillancourt

    license: (ISC) MIT

    About to write to C:\Users\vaill\Documents\work\WebProjects\ClevrBooks\server\package.json:

    {

        "name": "server",

        "version": "1.0.0",

        "description": "Backend Server for ClevrBooks app",

        "main": "src/app.js",

        "scripts": {

            "test": "echo \"Error: no test specified\" && exit 1"

        },

        "author": "Claude Vaillancourt",

        "license": "MIT"

    }

    Is this OK? (yes)

Then install ‘eslint’ as a development module

>npm install –save-dev eslint

Then insert the following commands to the “scripts” attribute of ‘package.json’:

“scripts”: {

“start”: “nodemon –verbose”,

“lint” “eslint \*\*/\*.js”,

“init”: “eslint –init”,

“test”: “echo \”Error: no test specified\” && exit 1”

},

We can now run the “init” script to initialize eslint.

>npm run init

The generated .eslintrc.json file should be like the following:

**{**

**"env": {**

**"browser": true,**

**"commonjs": true,**

**"es2021": true**

**},**

**"extends": "eslint:recommended",**

**"parserOptions": {**

**"ecmaVersion": "latest"**

**},**

**"rules": {**

**"indent": [**

**"error",**

**"space"**

**],**

**"linebreak-style": [**

**"error",**

**"windows"**

**],**

**"quotes": [**

**"error",**

**"single"**

**],**

**"semi": [**

**"error",**

**"never"**

**]**

**}**

**}**

Finally, create a .gitignore file with the following content:

**.DS\_Store**

**node\_modules**

**\*.sqlite**

**# Log files**

**npm-debug.log\***

**# Editor directories and files**

**.vscode**

**\*.suo**

**\*.ntvs\***

**\*.njsproj**

**\*.sln**

**\*.sw?**

Finally, create a subfolder ‘src’ containing the ‘app.js’ file that will be the main application code file. The file can only contain a console.log() for now.

We can now commit and push to GitHub.

## Database

Next step is to create a database user with phpMyAdmin and set the parameters in the C:\xampp\htdocs\clevrbooks\.env file.

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

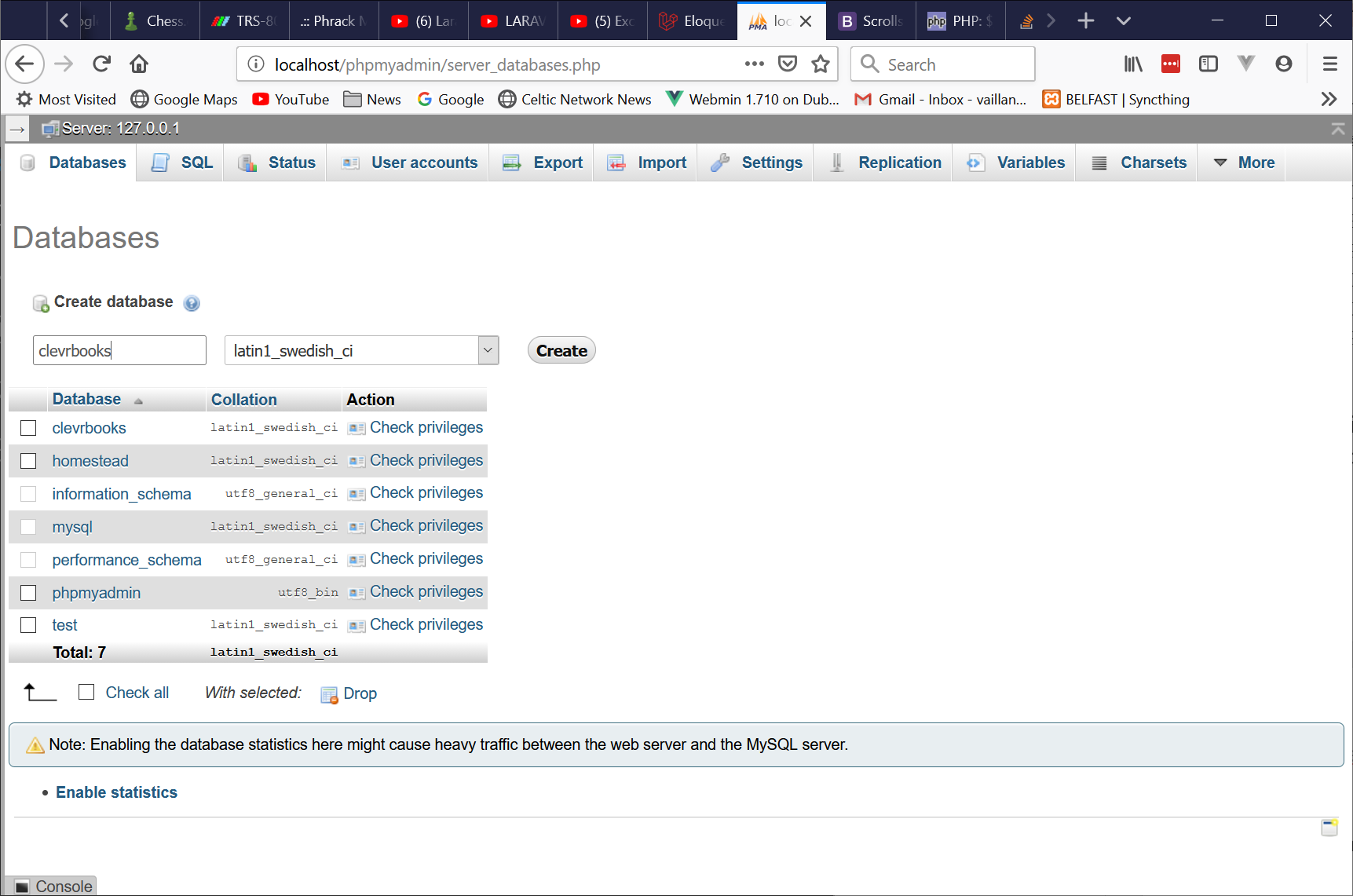
DB\_DATABASE=clevrbooks

DB\_USERNAME=clevrcode

DB\_PASSWORD=G5l4pDQZOI3Y

In this case, the password was generated by lastpass. The phpMyAdmin app has a generate button, but here, lastpass was used.

Then, from phpMyAdmin, create the clevrbooks database.



The collation field is Ok with the above default value. See [Character Sets and Collations in MySQL](https://dev.mysql.com/doc/refman/8.0/en/charset-mysql.html) for details.

### Create table migrations

1. CRUD stands for “Create/Read/Update/Delete”. [↑](#footnote-ref-1)
2. ORM stands for Object-Relational Mapping [↑](#footnote-ref-2)