REPO: https://github.com/VocabularyInReading/VIRS-Vocabulary-In-Reading

Clone or create the repo

Create a local MySQL account with credentials

username : root

password : root

‘mysql is not recognized as an internal or external command’

FIX: SET PATH=%PATH%;<your mysql bin folder location>

EXAMPLE: SET PATH=%PATH%;C:\Program Files\MySQL\MySQL Server 8.0\bin

Login via the MySQL CLI: mysql -u root -p

Enter the password “root” when prompted

Type the following command: create schema vir;

Exit: exit

Add tables: mysql -u root -p vir < /virs/Code/VIR-Backend/src/main/resources/schema.sql\

Add data to the tables: mysql -u root -p vir < /virs/Code/VIR-Backend/src/main/resources/data.sql

# VIR-Frontend

### Introduction

`VIR-Frontend App` built using the following:

- Developed using boostrap-v4.0.0-alpha.6

- ng-bootstrap

- Node.js: 14.15.4 (Latest versions might not work with the project.)

- angular-v4.3.1

- angular/cli-v1.3.2

- Official Angular i18n support.

- SCSS

- HTML5

### How to start

\*\*Note\*\* that this seed project requires \*\*node >=v6.9.0 and npm >=3\*\*.

\*\*INSTALL NODE IN ORDER TO BE ABLE TO USE 'npm' COMMANDS\*\*

In order to start the project use:

```bash

$ cd virs/Code/VIR-Frontend

# install the project's dependencies

$ npm install

$ npm uninstall -g @angular/cli

$ npm install -g @angular/cli@6.0.8

$ npm install puppeteer --save-dev

$ npm install xlsx (optional)

$ npm install file-saver (optional)

# watches your files and uses livereload by default run `npm start` for a dev server. Navigate to `http://localhost:4200/`. The app will automatically reload if you change any of the source files.

$ npm start

# prod build, will output the production application in `dist`

# the produced code can be deployed (rsynced) to a remote server

$ npm run build

```

Npm start will start the application in the front end if the backend is running as well the words from the database and backend functionality should be functional on port 4200

### Code scaffolding

Run `ng generate component component-name` to generate a new component. You can also use `ng generate directive/pipe/service/class/module`.

### Running unit tests

Run `ng test` to execute the unit tests via [Karma](https://karma-runner.github.io).

\*\*NOTE FOR WINDOWS USERS: this currently does not work properly for windows 7.

Works with out errors on Windows 10\*\*

### Running end-to-end tests

Run `ng e2e` to execute the end-to-end tests via [Protractor](http://www.protractortest.org/).

Before running the tests make sure you are serving the app via `ng serve`.

### Further help

(IF 'npm start' IS NOT WORKING, UNINSTALL & INSTALL NODE-SASS WITH 'npm uninstall node-sass' THEN 'npm install node-sass')

(IF 'ng' COMMANDS NOT WORKING, INSTALL ANGULAR/CLI WITH 'npm install -g @angular/cli')

To get more help on the Angular CLI use `ng help` or go check out the [Angular CLI README](https://github.com/angular/angular-cli/blob/master/README.md).

### Documentation //Might not be needed

- install it with npm :

```bash

$ npm i -g @compodoc/compodoc

```

- run it inside the folder of your Angular application

```bash

$ compodoc -p tsconfig.json

```

You can then run an HTTP server in that folder, or launch the previous command with the -s flag, and the tool will serve you the documentation, by default on http://localhost:8080

```bash

$ compodoc -p tsconfig.json -s

```

VIR-BACKEND

Make sure that JAVA is installed

Sometimes the project fails to virs.cmd run because system32 in the path is above java-version\bin so windows assigns the latest Java version. To fix this move JAVA\_HOME above system root by either dragging or using the arrow options.

For this installation you will need Java 1.8 or Java 8.

Set up an environmental variable called “APP\_ENCRYPTION\_PASSWORD” (the key is kept secret, the Product Owner should send you an Excel file with the passwords. As of now the app encryption password should be at the top of the excel file called Secret key for jasypt/ password)

Set up an environmental variable called “SENDGRID\_API\_KEY”. This is used for sending emails. NOTE: as of fall of 2022, the SENDGRID\_API\_KEY has been lost for over 3 years and we can not find a reason it was ever even needed.

Set up an environmental variable called “PROD” and set the value to 0. This is used by the application to determine whether or not emails can be sent (in production only, but this can be modified for debugging/testing.)

Install Tesseract for your system. <https://github.com/tesseract-ocr/tessdoc>

For development that doesn’t involve OCR or scanning images you can use Tesseract 5 but if you are working on OCR you may have to use Tesseract 4.

Create an environment variable TESSERACT\_PATH pointing to your installation directory. There the executable is.

Create an environment variable TESSDATA\_PREFIX pointing to the directory above the ‘tessdata’ one; should be in your installation directory but it can be downloaded from the web.

To Build:

./mvnw build (mvnw.cmd compile for Windows) (if ‘build’ doesn’t work use ‘compile’)

After the initial set up it is not necessary to run mvnw.cmd compile.

Virs.cmd install-only will compile and make changes to the source code so that the frontend and backend can communicate. When working on features that involve both the backend and frontend it is necessary to run virs.cmd install-only to test the whole application.

virs.cmd run will run the backend of the website

Prod:

Make sure there is a release and bundle folder

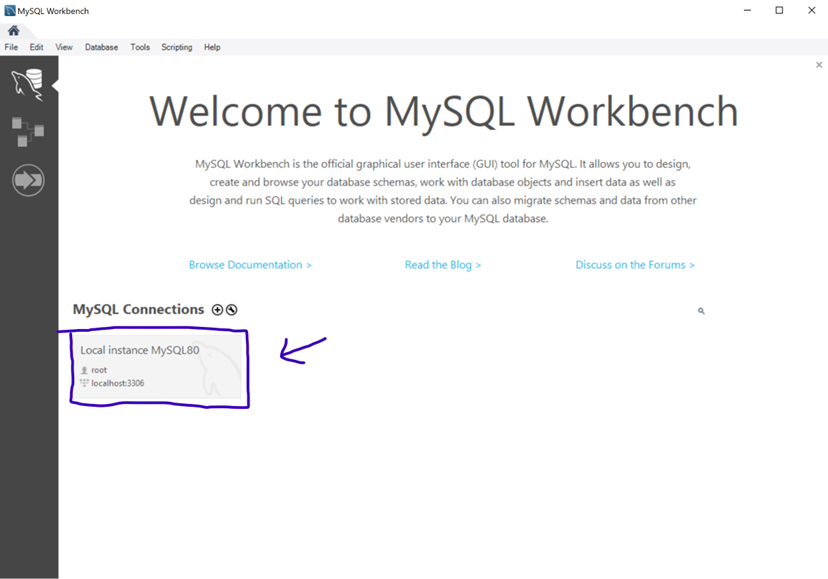
./virs.sh prod (‘virs.cmd prod’ for Windows)

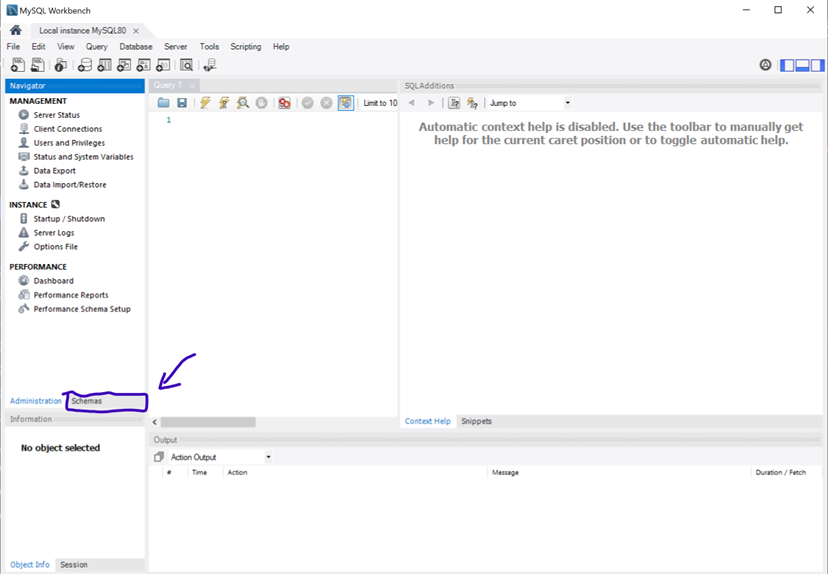
The zip file of the production will be located Code\VIR-Backend\release path and you can place that zip file in AWS Elastic Beanstalk to make it to the live website.

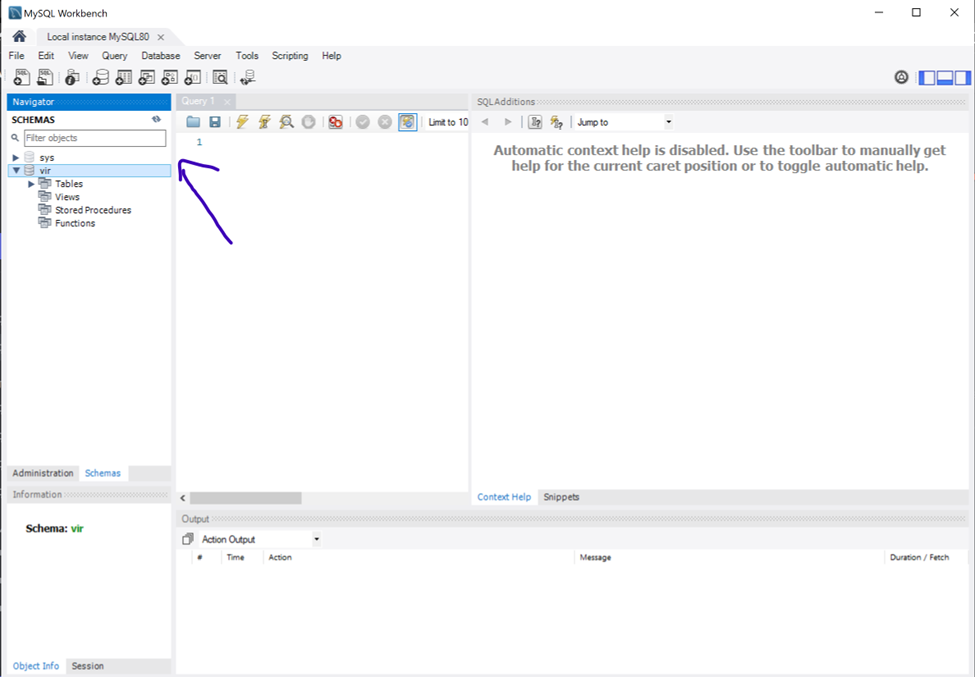
If testing errors arise:

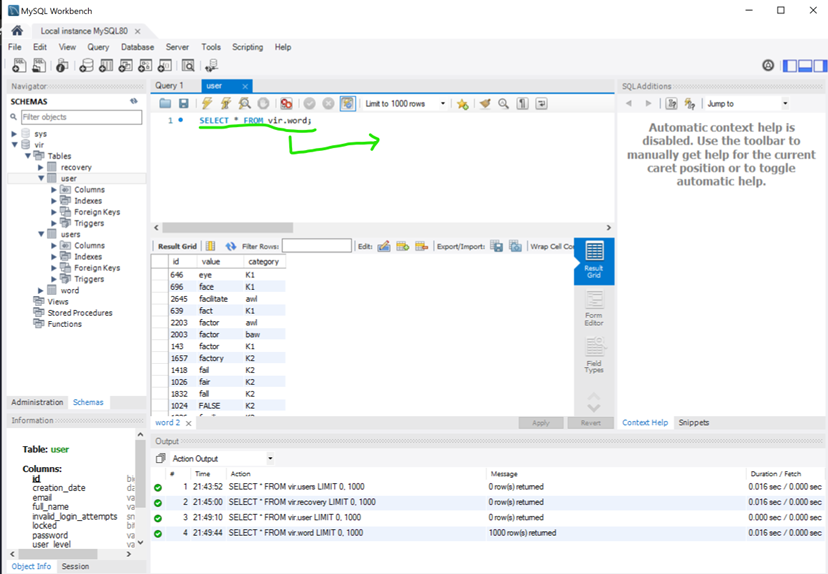
./virs.sh prod-build (‘virs.cmd prod-build’) for Windows)

##Additional Using MySQL workbench









Unlike other databases mysql doesn’t seem to have a way to view tables without running queries.