Messaging System

CSCI 3428

Installation and Maintenance Guide

Group 4

Contents

CSCI 3428 - Installation and Maintenance Guide

Group 4

Monday 18th November, 2019

1 User Access

The project will be hosted on the undergraduate computer science student's server at Saint Mary's University, and is publicly accessible at: ugdev.cs.smu.ca/~group4. The project has been developed and tested on the latest desktop versions of Chrome (v.78), Firefox (v.70), and Safari (v.13.0.1), though it should also be compatible with older versions and mobile platforms, provided there is support for HTML5, CSS3, and JavaScript.

On access, existing users will be prompted to log into the messaging system with their given account, while new users will have the option of creating their own account (providing they have a reference from an existing user). For testing purposes, the messaging system can be currently accessed using the following credentials:

Username: Bob
Password: 111

Authenticating new users, changing passwords, and adjusting any other user-related settings will be done by the administrator using the Django User-Management tools that are available on the server (see below).

2 Server Access

The project's files, databases, and server setup can be accessed via SSH:

Username: ________
Password: ________

Currently, all the tools and databases required for managing the project are installed on the server. This includes v.3.6.8 of Python3, which is used to install and manage the Django framework, and v.14.14 of MySQL. The project's database is accessible using:

Username: ________
Password: ________

The ReactJS framework was used to create the source files for the project's front-end, and while it is not required for the basic day-to-day running and administration of the system, any long-term changes to the project (including bug-fixes and adding functionality) will require that the server has a suitable JavaScript development environment available. The installation procedure for all of the above is detailed in the next section.

The current directory structure for the server is shown below. The web-facing public_html directory contains an optimized build version of the project, that condenses the various source and package files. While fully functional, these files are not (easily) readable, and are not intended to be modified directly. Changes should first be made to the source-code within the project directory, tested appropriately, and lastly built and included in public_html. Of particular importance are the contents of the messagingSystem/src/components/ directory, which includes the React and CSS front-end for the project, as well as the user-specific CSS files.

```
-
___ project-master/
     ___ Documentation/
     ___ LICENSE
     ___ README.md
     ___ messagingSystem/
          ___ assets/
          ___ components/
          ___ services/
          ___ src/
               ___ components/
                    ___ Communication.css
                    ___ Communication.js
                    ___ Login.css
                    ___ Login.js
                    ___ UserInterface.css
                    ___ UserInterface.js
               ___ index.css
               ___ index.js
               ___ router.js
          ___ utils/
___ public_html/
     ___ index.css
     ___ index.html
     ___ index.js
```

3 Migration and Maintenance

In order to migrate the project to a new server (or make and push any changes to the source-code), the host machine will require that Python3, MySQL, and a JavaScript development environment is installed. Ensure first that the Client-URL (curl) command-line utility for Linux is installed:

```
$ curl -V
```

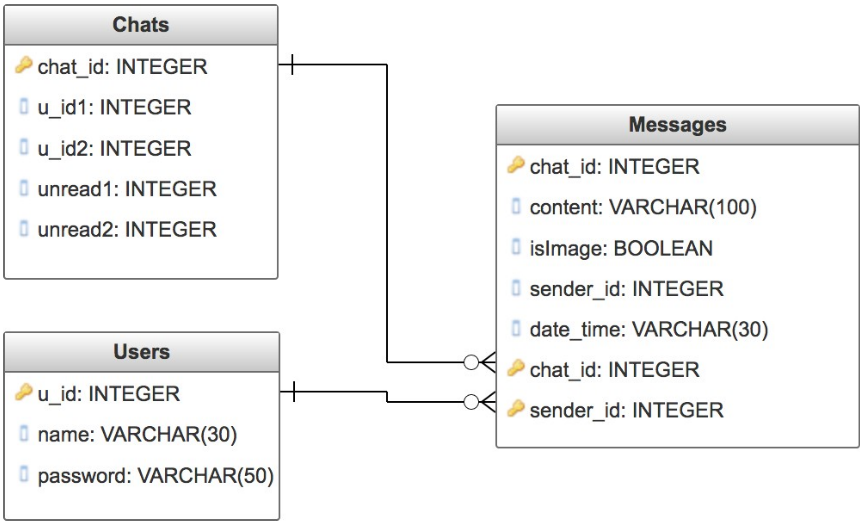
If not, install it using your system's appropriate package-installation tool. For Debian, Ubuntu, or related distributions, use:

```
$ sudo apt-get install curl
```

Most Linux distributions will include Python3 by default. Verify that it is installed, and install it if not. Preferably, select a distribution that includes the Python package manager pip, which will be used to install the Django framework. Otherwise, it can be obtained separately:

```
$ curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
$ python get-pip.py
$ pip install  Django
```

To configure the project's database, ensure that MySQL (v.14.14 or later) is installed on the host-machine, and structure it as follows:



In order to maintain the project's source code, a JavaScript development environment will need to be present on the host machine (in particular, the React framework). Ensure that the newest version of the Node.js runtime environment is installed. If not, install Node.js and additional build tools:

```
$ sudo apt-get install nodejs
$ sudo apt-get install build-essentials
```

As of v.8.10 of Node, and v.5.6 of NPM (bundled in the Node installation), a new React project can be created using:

```
$ npx create-react-app  my-app
$  cd  my-app
$  npm start
```

The current project can be run locally from within the messagingSystem directory by using NPM to first install missing dependencies (if needed) and start the application:

```
$ npm install
$ npm start
```