Administrationsanleitung

Büchersoftware

Inhaltsverzeichnis

1. Betriebssystem

Die Büchersoftware, nachfolgend App genannt, wurde für Linux programmiert und auf Ubuntu getestet.

Deswegen sollte man als Betriebssystem die LTS-Version von Ubuntu-Server nutzen. Dieses Betriebssystem installiert man dann in einem Docker, Virtuellen-Maschiene oder als Root-Server.

Die untenstehenden Befehle updaten ihr System auf den neusten Stand.

```
$  sudo apt update
```

```
$ sudo apt upgrade
```

2. Projekt herunterladen

2.1 „git“ Installieren

Um das Herunterladen zu vereinfachen, verwendet man „git“, welches mit folgendem Befehl installiert werden kann.

```
$  sudo apt install git
```

2.2 Dateien herunterladen

Nun verwendet man „git“ um das Projekt mit den aktuellen Updates herunter zuladen. Dies erledigt der untenstehende Befehl.

```
$ git clone https://github.com/lucakreativ/library-software.git
```

Um die folgenden Schritte zu vereinfachen, öffnet man den heruntergeladenen Ordner und führt dort die Befehle aus.

3. MySQL

3.1 Installation

Für die Installation benutzt man den unterstehenden Befehl.

```
$  sudo apt install mysql-server
```

3.2 Benutzer erstellen

Für die App wird ein MySQL-Benutzer benötigt.

Dafür loggt man sich in MySQL mit dem folgenden Befehl ein.

```
$  sudo mysql
```

Danach erstellt man einen Benutzer, dafür tippt man folgende Befehle ein.

Hinweis: Das rot markierte, verändert man zu seinen Parametern, wie z. B. Benutzername.

```
mysql>  CREATE USER 'newuser'@'localhost' IDENTIFIED BY 'password';
```

Um Rechte zu vergeben, benutzt man folgenden Befehl.

```
mysql>  GRANT ALL PRIVILEGES ON * . * TO 'newuser'@'localhost';
```

Diese müssen nun noch gespeichert werden.

```
mysql> FLUSH PRIVILEGES;
```

Die Änderungen wurden nun gespeichert.

3.3 Datenbank erstellen

Danach muss eine Datenbank erstellt werden. Um es zu vereinfachen, kann man die mitgelieferte Standard Bibliothek „bibliothek.sql“ als Grundbaustein nehmen. Dadurch spart man sich das erstellen von Tabellen.

Dafür erstellen man, in MySQL eingeloggt, eine Datenbank.

```
mysql>  CREATE DATABASE datenbank;
```

Dazu wechselt man zur erstellten Datenbank.

```
mysql> USE datenbank;
```

Danach benutzen man folgenden Befehl, damit die Standard-Datenbank geladen wird.

```
mysql> SOURCE bibliothek.sql;
```

Um MySQL zu verlassen, benutzt man `exit` .

3.4 Konfigurationsdatei ändern

In dem Ordner „App“ gibt es eine „config.ini“ Datei, diese dient als Konfigurations-Datei. In dieser Datei muss man „database“ zu dem vergebenen Namen der Datenbank ändern, den „user“ zum Benutzername von MySQL und „password“ zum vergebenen Passwort.

4. Authentifizierung mit Microsoft

Damit das Programm funktioniert, generiert man die API-Keys in Azure und fügt die Schlüssel in die „config.ini“ Datei hinzu.

5. Konfiguration der „App“

Für die Installation wird „pip“ benötigt, dafür führt man folgenden Befehl aus.

```
$ sudo apt install python3-pip
```

Zur einfachen Installation der librarys, wird „pip“ benötigt.

```
$  sudo pip3 install -r requirements.txt
```

Nun sind alle notwendige librarys installiert und man kann fortfahren.

6. Benutzer hinzufügen

Um einen Benutzer in der „App“ hinzuzufügen, führt man die Datei „add_user.py“ mit den Daten aus.

```
$ python3 add_user.py Benutzername E-Mail Passwort
```

Daraufhin sollte man eine Ausgabe auf dem Terminal bekommen, die so ähnlich aussieht wie unten, und in der letzten Zeile sollte „success“ stehen.

(Daten: Benutzername=Maria, Email=Maria@example.com Passwort=sicheresPasswort)

7. „App“ ausführen

7.1 Testen der „App“

Um die „App“ zu testen, führt man folgenden Befehl aus.

```
$ python3 app.py
```

Daraufhin sollte man diese Ausgabe bekommen.

```
* Serving Flask app 'app' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
```

Um zu testen, ob alles funktioniert, ruft man die IP-Aresse vom Server mit dem Port 5000 im Webbrowser auf. Daraufhin sollte man eine Login-Seite sehen.

7.2 Dauerhaftes ausführen der „App“

Um die „App“ auch im Hintergrund Ausführen zu können, benutzt man das vorinstallierte Programm „nohup“. Den folgenden Befehl ausführen, damit die „App“ im Hintergrund läuft.

```
$ nohup python3 app.py &
```

Nun muss man nur noch „Strg“ + „C“ drücken und alles läuft. Nun kann man sich auch ausloggen.

7.3 Beenden der „App“

Mit dem folgendem Befehl findet man die PID, Kurzform für Process ID, heraus.

```
$ ps -f -C "python3 app.py"
```

Danach benutzt man den folgenden Befehl „kill“ mit der zuvor herausgefundenen PID, um den Prozess zum Beenden zu bringen.

```
$ kill PID
```

7.4 Log-Dateien

Standardmäßig werden die Log-Daten in der Datei „log.log“ gespeichert. Diese kann man mit beliebigem Editor anschauen.

8 Updates herunterladen

Die Updates können auch mit „git“ sehr einfach heruntergeladen werden. Dafür öffnet man den Ordner „library-software“ und nutzt folgende Befehle.

```
$ git stash
```

```
$ git pull
```

```
$ git stash pop
```

```
$ git checkout –theirs config.ini
```

Nun muss man nur noch die „App“ neu starten und alle Updates werden angewendet.