**Video Analytics for PPE Detection**

**- A PPE Detection System -**

**Beta Version**

Installation Manual



Sponsored by TIEFA

# Overview

This web application titled **‘Video**​ **Analytics for Personal Protective Equipment’,** uses​ video analytics and Artificial Intelligence to detect Personal Protective Equipment worn by visitors’ to the ECTC pilot plants, and assess their compliance with PPE safety regulations. In the event of a violation, the result will be logged in the database and the Safety Manager notified of the violation through Slack and the dashboard.

This document provides instructions for the installation of this web application.

# Table of Contents

[**Overview 1**](#_Toc94924260)

[**Table of Contents 2**](#_Toc94924261)

[**Prerequisites 4**](#_Toc94924262)

[Setup Folder Contents 4](#_Toc94924263)

[Installing Required Services 11](#_Toc94924264)

[MongoDB Community Server 11](#_Toc94924265)

[MongoDB Compass (Optional) 12](#_Toc94924266)

[Visual Studio Build Tools 12](#_Toc94924267)

[Apache 2.4.46 & Visual Studio C++ binaries 13](#_Toc94924268)

[Python 3.7.7 13](#_Toc94924269)

[**Installation 15**](#_Toc94924270)

[Setup Database 15](#_Toc94924271)

[Deploy application on server 17](#_Toc94924272)

[Setup Apache service 17](#_Toc94924273)

[Install python packages from requirements.txt 18](#_Toc94924274)

[Configure httpd.conf 19](#_Toc94924275)

[Start Apache service 20](#_Toc94924276)

[Deployment Completed 21](#_Toc94924277)

[**Troubleshooting 21**](#_Toc94924278)

[Cannot find screen to enter connection details 21](#_Toc94924279)

[Connection unsuccessful 22](#_Toc94924280)

[Unable to install mod\_wsgi during “pip install -r requirements.txt” 23](#_Toc94924281)

[Unable to start apache service due to permission 23](#_Toc94924282)

[500 internal server error on apache 24](#_Toc94924283)

[**Download Installation Tools (Optional) 25**](#_Toc94924284)

[MongoDB Community Server 25](#_Toc94924285)

[MongoDB Compass 25](#_Toc94924286)

[Visual Studio Build Tools 25](#_Toc94924287)

[Apache 2.4.46 & Visual Studio C++ binaries 25](#_Toc94924288)

[Python 3.7.7 26](#_Toc94924289)

[**Updated Folder Contents (28 Jan 2022) 27**](#_Toc94924290)

[**Replacing MongoDB with PostgreSQL 28**](#_Toc94924291)

[Download PostgreSQL 28](#_Toc94924292)

[Files needed 28](#_Toc94924293)

[Setting up database with pgAdmin 29](#_Toc94924294)

[Edit backend files 35](#_Toc94924295)

[Model Evaluation Using mAP 38](#_Toc94924296)

# Prerequisites

**-- Ensure that you have administrator privileges before proceeding --**

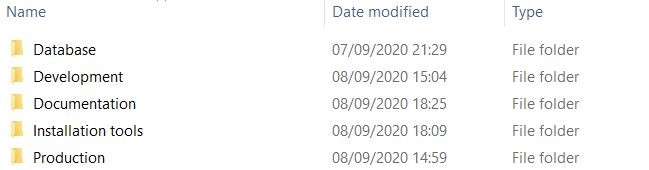
**-- Please use 7-Zip or WinRAR to extract the files inside the Setup.zip file --**

7-Zip: [https://www.7-zip.org/a/7z1900-x64.ex](https://www.7-zip.org/a/7z1900-x64.exe)​ [e](https://www.7-zip.org/a/7z1900-x64.exe)

WinRAR: [https://www.win-rar.com/postdownload.html?&L=](https://www.win-rar.com/postdownload.html?&L=0)​ [0](https://www.win-rar.com/postdownload.html?&L=0)

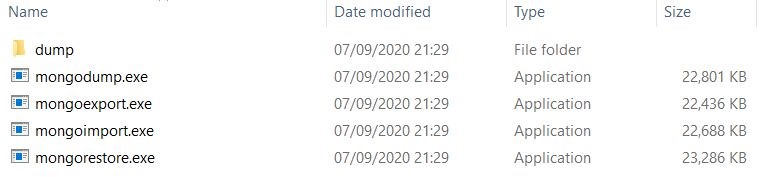
## Setup Folder Contents

The Setup folder should contain the following folders:



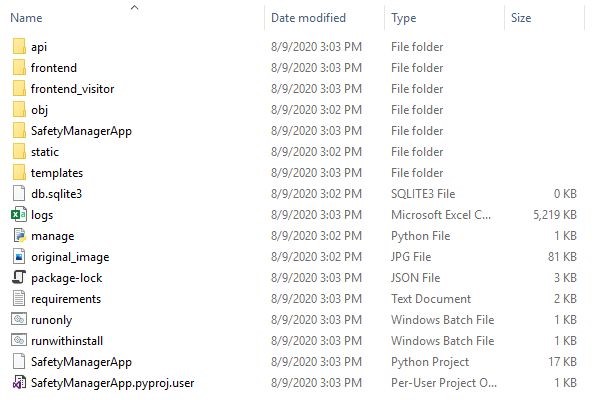
* **Database:** ​Contains the executable files and table creation files for database setup
* **Development:** ​Contains the project’s source code and files (for developers)
* **Documentation:** ​Contains the documentation for the application to assist future developers and users of the application (User Manual, Program API, etc.)
* **Installation tools:**​ Contains the tools to install the services required for the application **● Production:** ​Contains the project’s source code and files (for deployment)

**Database**​ folder contains the following files:​



* **dump:** Contains the json files of the tables in the SafetyManagerApp​
* **mongodump.exe:** An executable file that exports all the data in the local mongoDB​ server into JSON folder (dump)
* **mongoexport.exe:** An executable file that exports specific database in the local​ mongoDB server into JSON folder (dump)
* **mongoimport.exe:** An executable file that imports specific collection of a database in​ the local mongoDB server from a folder of JSON files with the default folder (dump)
* **mongorestore.exe:** An executable file that restores the entire database in the local​ mongoDB server from a folder of JSON files with the default folder (dump) **Documentation**​ folder contains the following files:​

Files inside ThisCode Folder



## Installing Required Services

Open up the **Installation**​ **Tools** folder​ located in the Setup.zip provided. If you encounter any issues with the installation files, please download them using the links provided under the

**Download Installation Tools** section.​

### MongoDB Community Server

1. Run the **mongodb-windows-server.msi**​ installer​
2. Follow the MongoDB Community Server installation wizard and ensure that the options selected follow those shown below.

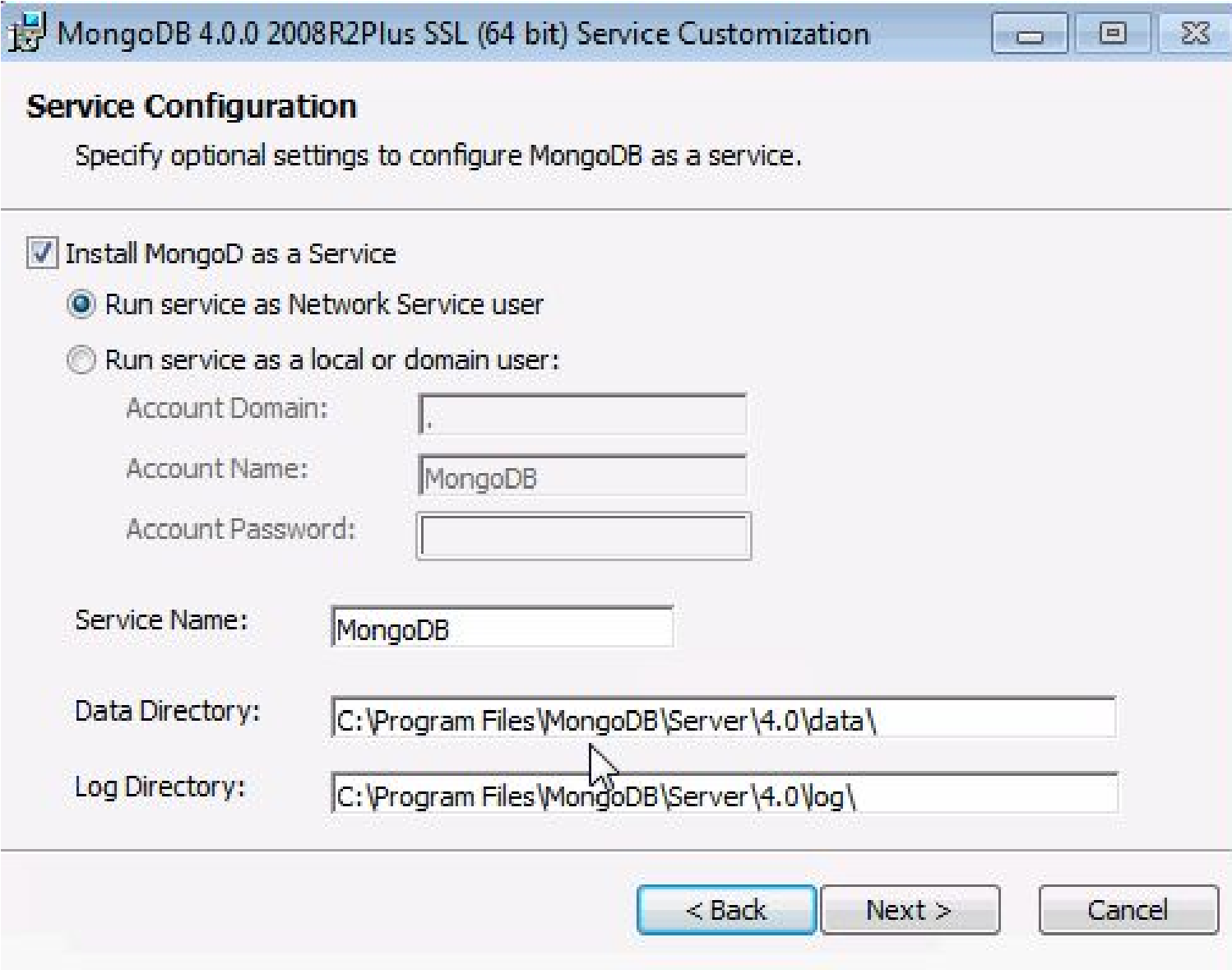
○ **Choose setup type** : Complete​

○ **Install MongoDB as a service** : Checked​

○ **Run service as a Network Service user** :​ Checked

○ **Service Name** : MongoDB​

**○ Install MongoDB Compass :** Checked​



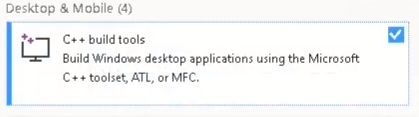
### MongoDB Compass (Optional)

If ​**MongoDB Compass cannot be found**​ after installing the ​**MongoDB Community Server**​, follow the steps below to install it.

1. Run ​**mongodb-compass-1.22.1-win32-x64.msi** ​installer
2. Follow the installation wizard

### Visual Studio Build Tools

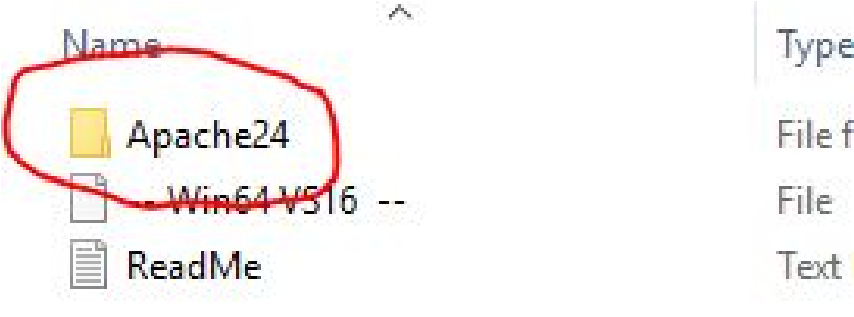
1. Run ​**Vs\_buildtools\_\_2082216191.1549097230.exe**
2. Under ​**Workloads**​, Select ​**C++ build tools**



1. Click ​**Install**

### Apache 2.4.46 & Visual Studio C++ binaries

1. Extract the ​**httpd-2.4.46-win64-VS16.zip**​ and move the “Apache24” folder found inside onto your C drive (e.g. ​**C:/Apache24**)​



1. Run ​**VC\_redist.x64.exe** ​and install “Microsoft Visual C++ 2015-2019 Redistributable

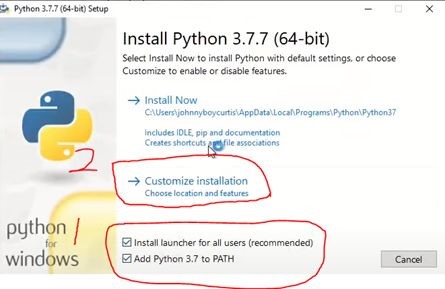
(x64)”

### Python 3.7.7

1. Run **Python-3.7.7-amd64.exe**​
2. Ensure that the following are selected
   1. Install launcher for all users (recommended)

○ Add Python 3.7 to PATH

1. Click **“Customize installation”**​



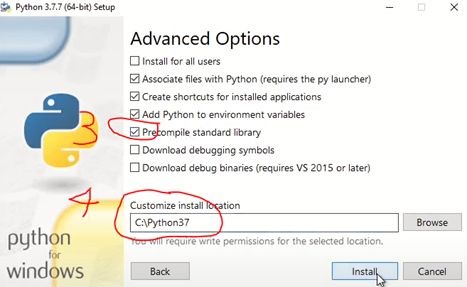
1. Check that the following options are selected
   1. Associate files with Python (requires the py launcher)

○ Create shortcuts for installed applications

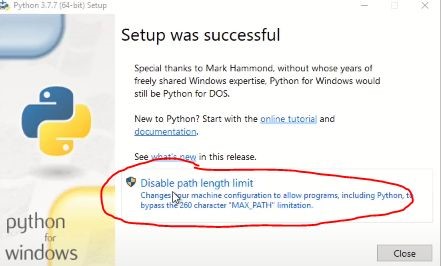
○ Add Python to environment variables

○ Precompile standard library

1. Change the install location to your C drive (e.g. ​**C:\Python37**​)



1. Click ​**Install**
2. After the installation is completed, click “​**Disable path length limit”**



# Installation

## Setup Database

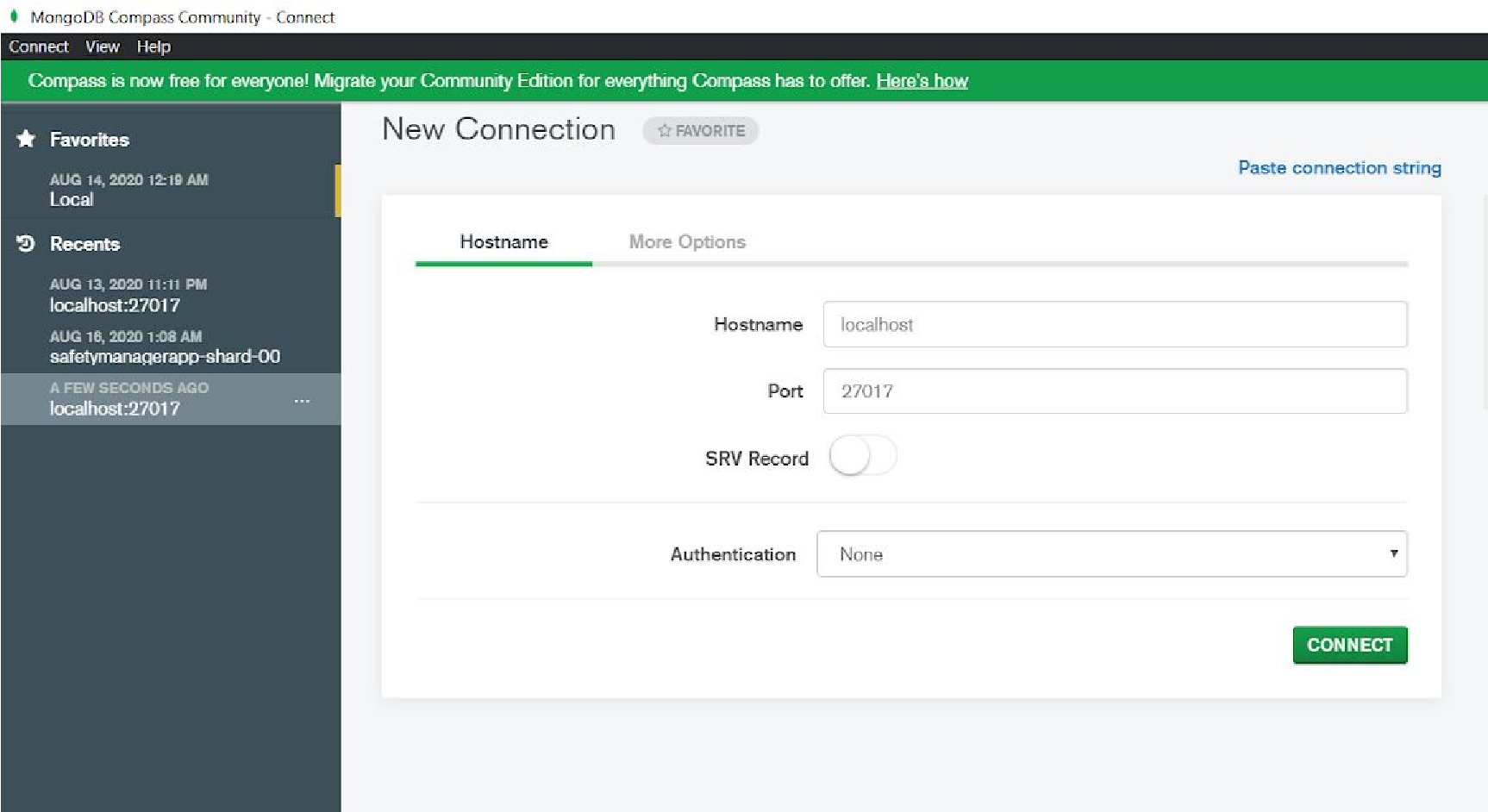
1. Open MongoDB Compass
2. Under New Connection, enter the following details in the respective fields

○ **Hostname** ​: localhost

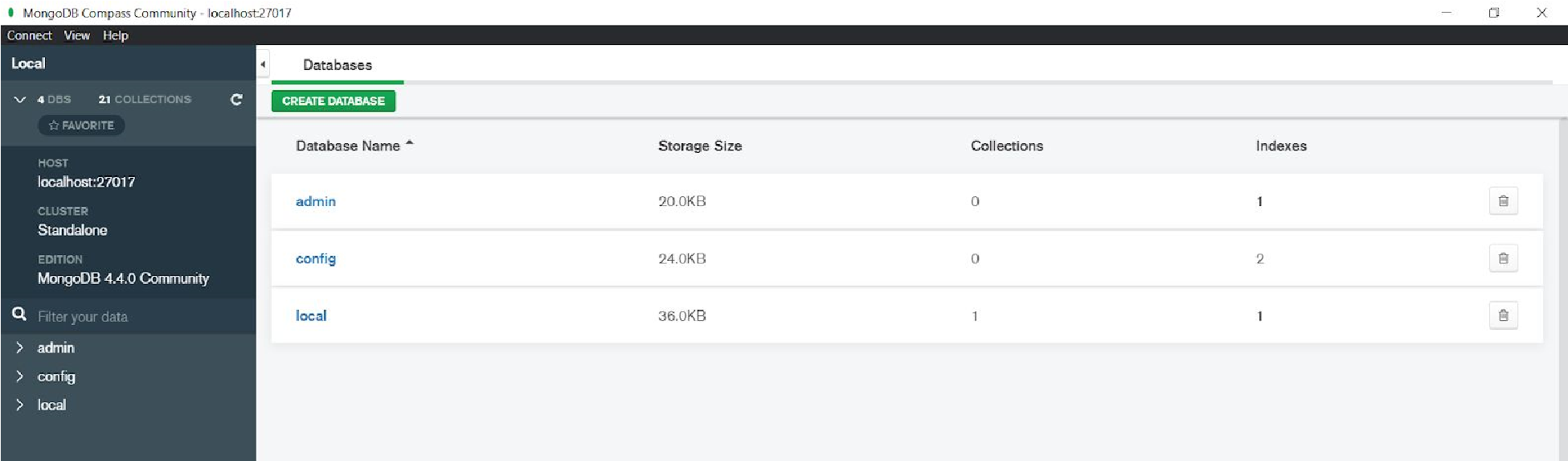
○ **Port** ​: 27017

○ **SRV Record**​ : unchecked

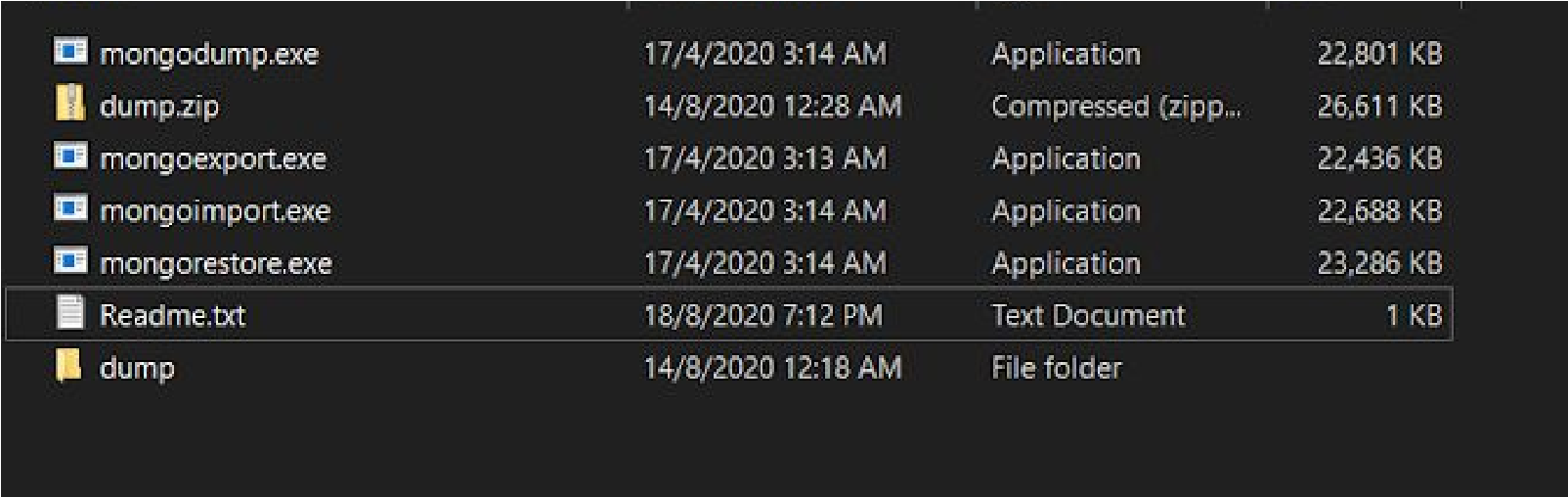
○ **Authentication** ​: None



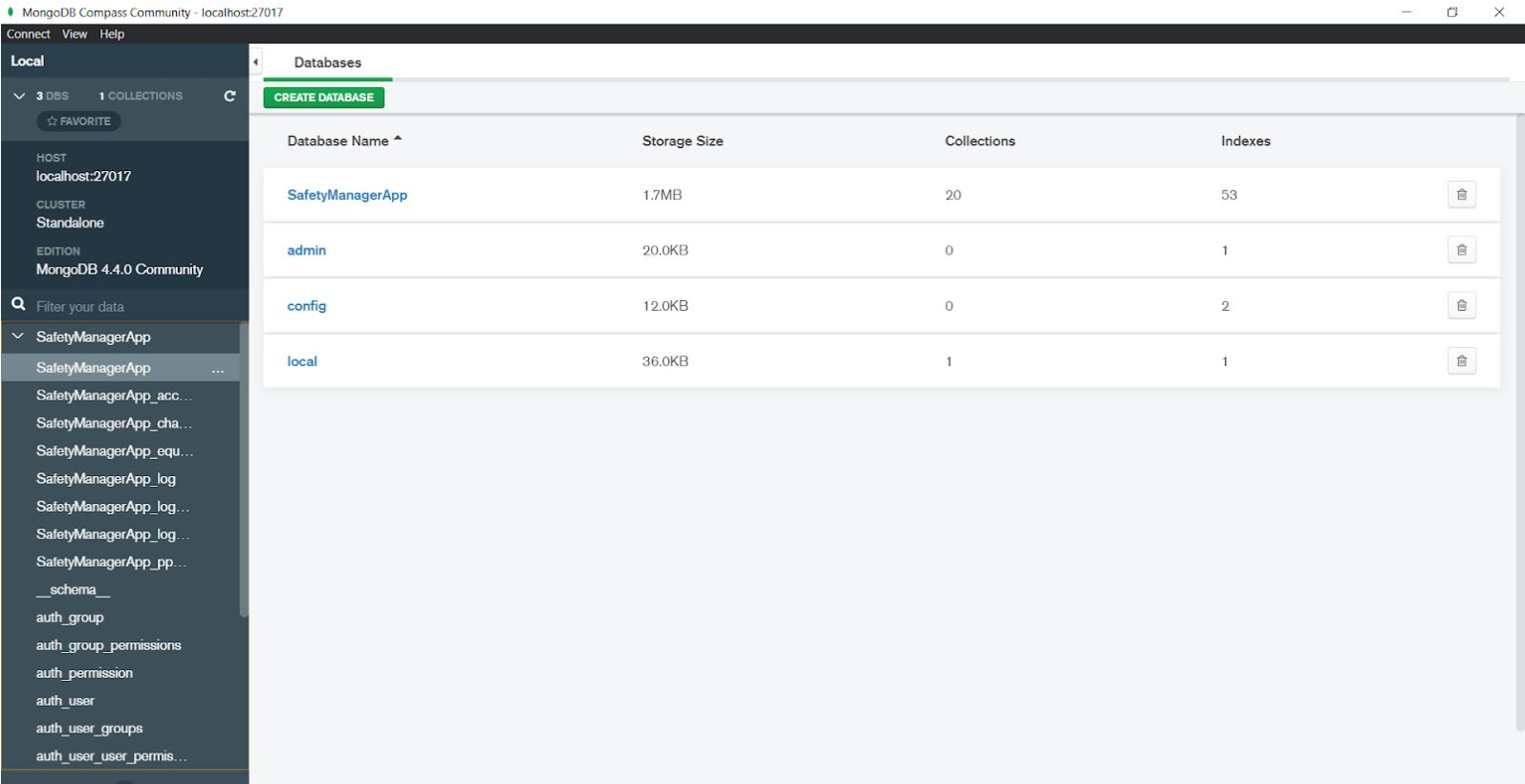
1. The following screen will appear if the connection was successful. Please refer to the **Troubleshooting** ​section of this document if the screen below is not shown.



1. Open up the **Database**​ folder. The folder should contain the following files:​



1. Run **mongorestore.exe**​
2. Refresh MongoDB Compass using the refresh icon and a new “SafetyManagerApp” database will appear indicating its successful restoration.



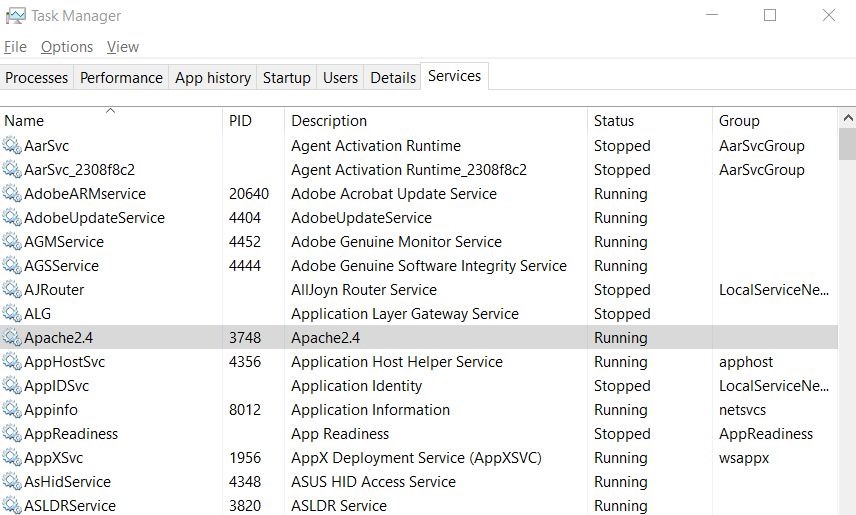
1. If the restoration of the database is successful, please proceed to the next section of this document

## Deploy application on server

1. Open the ​**“Production”**​ folder
2. Copy the ​**“DIT\_FYP\_APPE”**​ folder onto the desktop or your preferred location

### Setup Apache service

1. Run ​**Command Prompt** ​as Administrator
2. Enter “​**cd C:/Apache24/bin**​”
3. Enter “​**httpd.exe -k install**​”
4. Enter “​**httpd.exe -k start**​”
5. Open ​**Task Manager**​ and navigate to ​**services → Apache2.4**
6. Under the ​**“Status”**​ column, check that it says “​**Running”**



1. Open your browser (Microsoft Edge, Google Chrome, etc.) and enter **localhost**​ in the address bar. If the following message is displayed, you have set up the Apache service successfully:



### Install python packages from requirements.txt

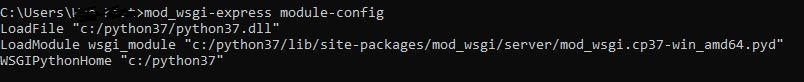
1. Run **Command Prompt**​
2. Enter “**cd** ​ **<**​ **path>/DIT\_FYP\_APPE/SafetyManagerApp**​ ”​

○ **Replace <path>** with the path of the location you placed the ​ **DIT\_FYP\_APPE**​ folder in (e.g. **C:/Users/MyUser/Desktop/**​ DIT\_FYP\_APPE/SafetyManagerApp)​

1. Enter “**pip install -r requirements.txt**​ ”.​
2. Enter “**pip install -r requirements.txt**​ ” again as some packages will not be installed on​ the first installation.

### Configure httpd.conf

1. Run **Command Prompt**​
2. Enter “**mod\_wsgi-express module-config**​ ”. The output should be similar to this:​



1. **Copy** the output​
2. Open **httpd.conf**​ using a text editor. This can be found in ​ **Apache24/conf/httpd.conf**​
3. Insert the following at the bottom of the file and make the necessary changes required (those that are highlighted):

● **Replace all <DIT\_FYP\_APPE folder path>**​ with the path of the location you placed the ​**DIT\_FYP\_APPE**​ folder in

(e.g. ​**C:/Users/MyUser/Desktop/**​DIT\_FYP\_APPE)

|  |
| --- |
| ServerName localhost:80    # Django Project  # Paste the output of module-config here  WSGIScriptAlias / "​<DIT\_FYP\_APPE folder  path>​/DIT\_FYP\_APPE/SafetyManagerApp/SafetyManagerApp/wsgi.py"  WSGIPythonPath "​<DIT\_FYP\_APPE folder path>/DIT\_FYP\_APPE/SafetyManagerApp"​  WSGIApplicationGroup %{GLOBAL}    <Directory "​<DIT\_FYP\_APPE folder  path>​/DIT\_FYP\_APPE/SafetyManagerApp/SafetyManagerApp/">  <Files wsgi.py>  Require all granted  </Files> </Directory>  Alias /static "​<DIT\_FYP\_APPE folder path>/DIT\_FYP\_APPE/SafetyManagerApp/static/"​ <Directory "​<DIT\_FYP\_APPE folder path>/DIT\_FYP\_APPE/SafetyManagerApp/static/">​ Require all granted  </Directory>    Alias /visitor/static "​<DIT\_FYP\_APPE folder path>​/DIT\_FYP\_APPE/SafetyManagerApp/static/"  <Directory "​<DIT\_FYP\_APPE folder path>/DIT\_FYP\_APPE/SafetyManagerApp/static/">​ Require all granted </Directory> |

### Start Apache service

There are two ways you can start the service (Must have administrator privileges)

1. **First Way (recommended):** 
   1. Open **Task Manager**​
   2. Navigate to **services**​
   3. Right click **Apache2.4**​
   4. Click **start**​
2. **Second Way:** 
   1. Open **Command Prompt**​
   2. Enter cd “**C:/Apache24/bin**​ ”​

* 1. Enter “**httpd.exe -k start**​ ”​

### Deployment Completed

Open a browser and enter “**localhost**​ ​” in the address bar to view it.

Note:

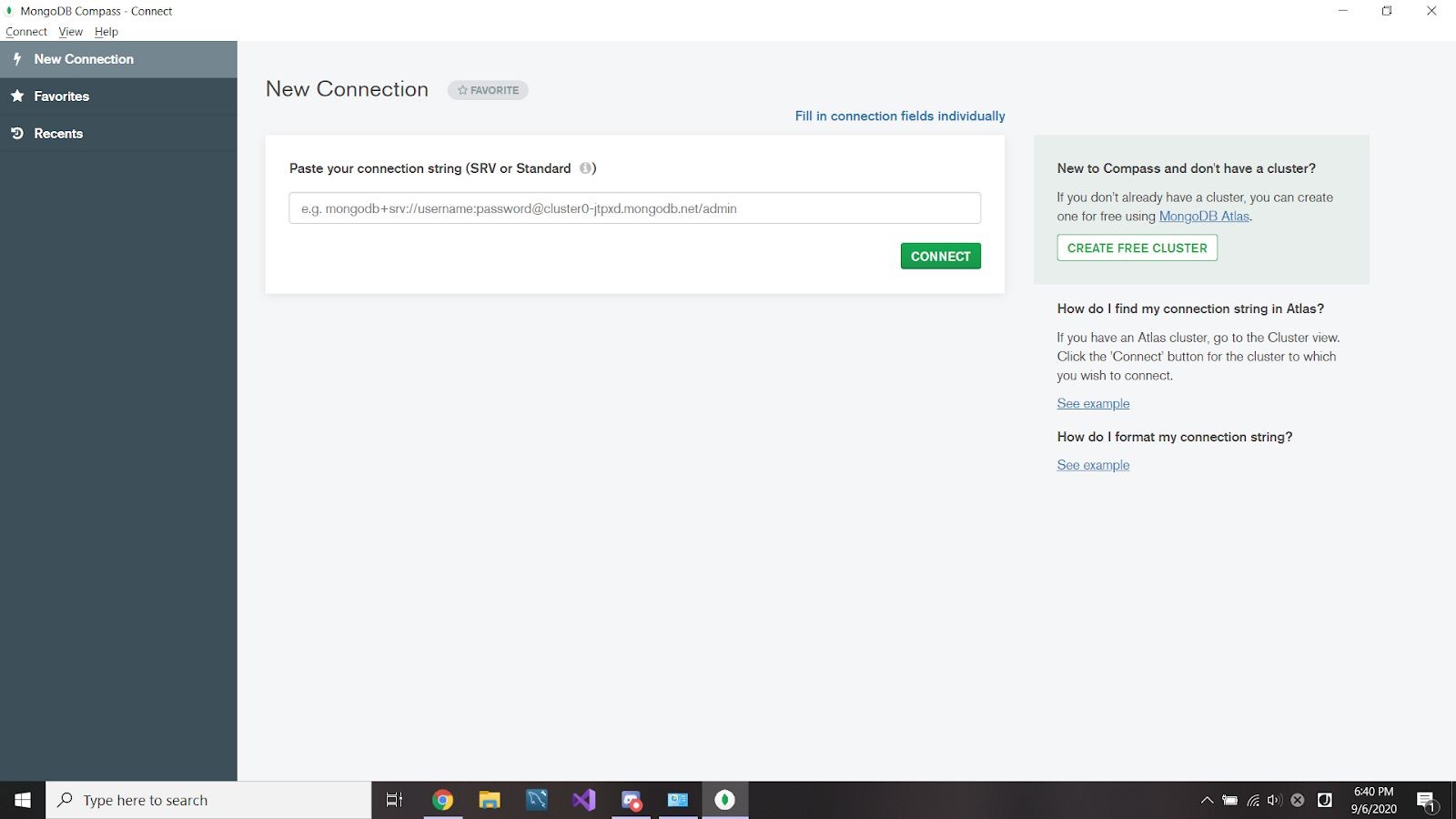
* This application is designed to run in Google Chrome or Microsoft Edge.
* Avoid using Internet Explorer or other non-tested browsers (Opera, etc.) as there may be some browser compatibility issues with the code used in the application.
* Avoid using Mozilla Firefox as it does not support the text-to-speech functionality.

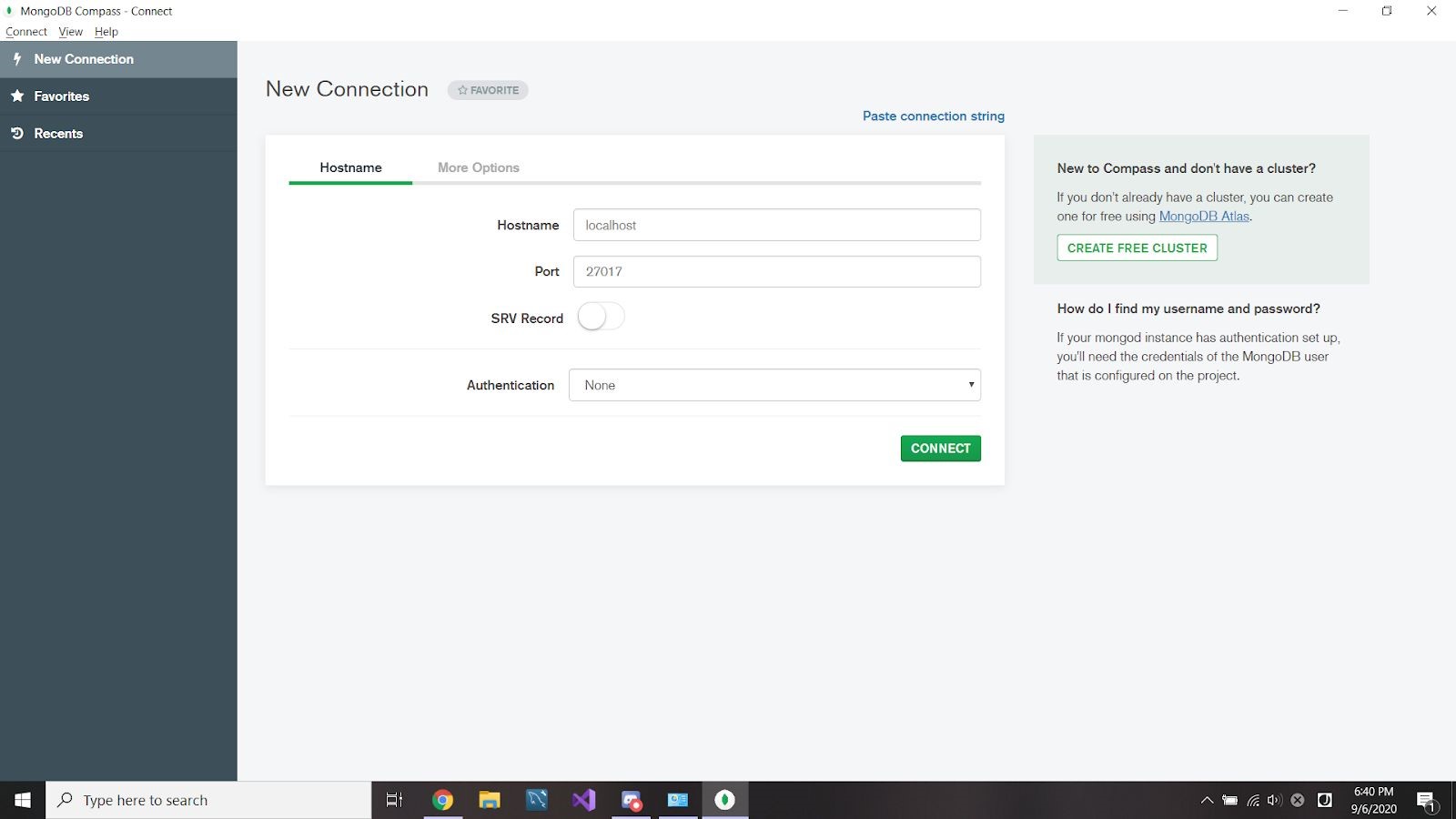
--------------------------------------------- **END OF INSTALLATION**​ ---------------------------------------------​

# Troubleshooting

## Cannot find screen to enter connection details

Go to New Connection → Fill in connection fields individually





## Connection unsuccessful

Change the authentication to Username / Password and then fill in the following details.

* **Hostname** :​ localhost
* **Port** : 27017​
* **SRV Record** : unchecked​
* **Authentication** :​ Username / Password
* **Username** :​ admin
* **Password** :​ admin (mongoDB default password)

**\*\* You are strongly advised to change the admin password for security purposes**

## Unable to install mod\_wsgi during “pip install -r requirements.txt”

Visual Studio Build Tools is required before you install mod\_wsgi.

## Unable to start apache service due to permission

Administratives rights are required before you are allowed to start the service. Please use an admin account.

## 500 internal server error on apache

In the “SafetyManagerApp” folder (containing Django backend files), look for settings.py and under the “LOGGING” configuration settings, change the highlighted portion of the settings to the appropriate file path where the “file.log” is located. The “file.log” file should be under the folder containing the application code. Note that the “file.log” may not appear as a .log file, and can be a text document.Text

Description automatically generated

# Download Installation Tools (Optional)

**-- Installation Tools required are provided in the Setup.zip --**

**-- Alternatively, you may download the tools using the links below --**

## MongoDB Community Server

* **Version 4.4.0 (msi)** <https://fastdl.mongodb.org/windows/mongodb-windows-x86_64-4.4.0-signed.msi>
* **Latest**

<https://www.mongodb.com/try/download/community>

## MongoDB Compass

* **Version 1.22.1 (msi)** <https://downloads.mongodb.com/compass/mongodb-compass-1.22.1-win32-x64.msi>
* **Latest**

<https://www.mongodb.com/try/download/compass>

## Visual Studio Build Tools

<https://visualstudio.microsoft.com/thank-you-downloading-visual-studio/?sku=BuildTools&rel=16>

## Apache 2.4.46 & Visual Studio C++ binaries

* **Apache 2.4.46**

<https://www.apachelounge.com/download/VS16/binaries/httpd-2.4.46-win64-VS16.zip>

* **14.26.28720.3 Visual C++ Redistributable for Visual Studio 2015-2019**

<https://aka.ms/vs/16/release/VC_redist.x64.exe>

## Python 3.7.7

<https://www.python.org/ftp/python/3.7.7/python-3.7.7-amd64.exe>

# Updated Folder Contents (28 Jan 2022)

Note that although the file contents have been updated, the set-up and installation is the same as before, so do follow the steps for installation as stated under the **Prerequisites** and **Installation** sections.

Under “thisCode” folder should contain these following files and folders:

A screenshot of a computer

Description automatically generated with medium confidence

**api** Folder: Contains the files for the various API and serializers used in the web app.

**Database** Folder: Contains the MongoDB set-up files. To set up MongoDB, steps are the same as under “Setup Database”.

**frontend** and **frontend\_visitor:** contains the files used and source code for developers to work on to make changes to the application.

**static, templates and remaining files:** These files and folders are used to render the Vue.js framework code in and other assets are used for various functionalities for the app.

# Replacing MongoDB with PostgreSQL

## Download PostgreSQL

https://www.postgresql.org/download/

Graphical user interface, text, application

Description automatically generated

During installation, make sure that the following components “**PostgreSQL Server**”, “**pgAdmin 4**”, “**Command Line Tools**” are ticked so that they are installed. Proceed with default set-up in every step of the installation.

## Files needed

You will need the following files/folders: **TableCreation.txt** and the folder **Database CSVs**.

## Setting up database with pgAdmin

1. Start up **pgAdmin** and set the master password.

Note: When pgAdmin is first started up, a master password has to be set to access the service. This master password will be used when starting up pgAdmin in the future.

Graphical user interface, text, application, email

Description automatically generated

2. Click on **Add New Server**

Graphical user interface, application

Description automatically generated

3. Under “General”, set the **Name** of the server.

Graphical user interface, application

Description automatically generated

4. Under “Connection”, set the **Host name/address** to “localhost” and **Password** (for the default user “postgres”). Click on the **Save** button below.

Note: the password set for the user “postgres” will be used for subsequent connections to the database server.

Graphical user interface, application

Description automatically generated

5. You will automatically be connected to the server after saving the server settings. Right-click on the server you just created (in this case, mine is “example”). Click on “Databases” > Create > Database…

Graphical user interface, application

Description automatically generated

6. Set the database name to “SafetyManagerApp” and click the **Save** button.

Graphical user interface, application

Description automatically generated

7. After the database has been created, right-click on “SafetyManagerApp” and select **Query Tool**.

Graphical user interface, application

Description automatically generated

8. Copy and paste the commands in TableCreation.txt into **Query Editor** and run the commands. (F5)

Graphical user interface, text, application, email

Description automatically generated

9. After the tables have been created, on the left sidebar, you can see that the tables have been created. Right-click on any of the tables that start with “SafetyManagerApp\_” and select “Import/Export”.

A picture containing table

Description automatically generated Graphical user interface, text, application

Description automatically generated

10. Under **Import/Export**, make sure that it is set to “Import”. Set the filename as the complete filepath of where the directory **Database CSVs** is located, and add the relevant csv filename at the back. e.g If table selected is “SafetyManagerApp\_accesstoken”, the filename should be “…\Database CSVs\accesstoken.csv”. Set **Format** to “csv”. Under **Header**, set to “Yes” and **Delimiter** should be set to a comma, “,”. Click on the **OK** button below.

Repeat this step until the data has been imported to the following tables: **SafetyManagerApp\_accesstoken**, **SafetyManagerApp\_channel**, **SafetyManagerApp\_equipment**, **SafetyManagerApp\_log** and **SafetyManagerApp\_ppeselection**.

Graphical user interface, application, Teams

Description automatically generated

## Edit backend files

1. In **thisCode/SafetyManagerApp/settings.py**, replace the database settings as such:

Text

Description automatically generated

Note: the **‘PASSWORD’** field is the value of the password given in step 4 of “Setting up database with pgAdmin”.

|  |
| --- |
| DATABASES = {  'default' : {  'ENGINE': 'django.db.backends.postgresql\_psycopg2',  'NAME': 'SafetyManagerApp',  'USER': 'postgres',  'PASSWORD': 'password',  'HOST': 'localhost',  }  } |

2. In **thisCode/SafetyManagerApp/models.py**, replace the models as such:

|  |
| --- |
| from django.db import models  from django.contrib.postgres.fields import ArrayField  from django import forms  from django.contrib import admin  from django.db.models.fields import CharField  from SafetyManagerApp import settings  class equipment(models.Model):  equipmentName = models.CharField(max\_length=100)  equipmentIcon = models.CharField(max\_length=100)  objects = models.Manager()    class log(models.Model):  image = models.TextField()  time = models.DateTimeField()  objectsDetected= ArrayField(models.CharField(max\_length=100, blank=True))  objectsViolated = ArrayField(models.CharField(max\_length=100, blank=True))  objects = models.Manager()  class accesstoken(models.Model):  tokenStr = models.CharField(max\_length=100)  isValid = models.BooleanField(default=True)  objects = models.Manager()  class channel(models.Model):  channelStr = models.CharField(max\_length=100)  isValid = models.BooleanField(default=True)  tokenStr = models.CharField( max\_length=100)  objects = models.Manager()  class ppeselection(models.Model):  timestamp = models.DateTimeField(auto\_now=True)  selectionEquipment = ArrayField(models.CharField(max\_length=100, blank=True))  objects = models.Manager()  class details:  pass  class detectionObject:  pass  class equipmentList(object):  def \_\_init\_\_(self):  self.equipmentList = equipment.objects.values()  def equipmentDict(self):  equipmentDictionary = {}  for dict in self.equipmentList:  id = dict.get('id')  name = dict.get('equipmentName')  equipmentDictionary.update({id : name})  return equipmentDictionary  def equipmentIdDict(self):  equipmentidDictionary = {}  for dict in self.equipmentList:  id = dict.get('id')  name = dict.get('equipmentName')  equipmentidDictionary.update({name : id})  return equipmentidDictionary |

3. In **thisCode/api/ppeselection/serializers.py**, replace the serializer as such:

|  |
| --- |
| class PPESelectSerializer(serializers.Serializer):  id = serializers.IntegerField()  timestamp = serializers.DateTimeField()  selectionEquipment = serializers.ListField(child = serializers.IntegerField()) |

4. Restart Apache server and launch the application to make sure that it is working the same as before.

# Model Evaluation Using mAP

All files needed for calculation of the mAP score are under “calculate mAP.zip”. Instructions are also included in the zip folder, written in “instructions.txt”.