# User Guide

Contents

[User Guide 1](#_Toc93324819)

[Introduction 2](#_Toc93324820)

[Installation and Using the Application 2](#_Toc93324821)

[Download and install requisite tools: 2](#_Toc93324822)

[Install and run the Wireguard VPN Server (Recommend) 3](#_Toc93324823)

[Install and run the Database System 3](#_Toc93324824)

[Launch the API Backend System 4](#_Toc93324825)

[Launch the Web Application 4](#_Toc93324826)

[Launch the Mobile Application (Optional) 5](#_Toc93324827)

[Setup Nginx for Reverse Proxying (Production) 5](#_Toc93324828)

[Submit a corruption incident through DOSSA, the Mobile Application 6](#_Toc93324829)

[Access the Dashboard as an Analyst 7](#_Toc93324830)

[Access the Institutions List as an Analyst 8](#_Toc93324831)

[Access the Reporting as An Analyst 8](#_Toc93324832)

## Introduction

This guide intends to show users how to install, create an incident through the Dossa mobile application, and access other functions of the Dossier Management System web application. The following system design in Figure 1 shows the Dossier Management System as a collection of microservices working together.

Graphical user interface

Description automatically generated

Figure 1. System design of communication flow between users and backend services

## Installation and Using the Application

The Dossier Management System consists of five subsystems: VPN Server using Wireguard, the database system, the backend, a web application, and a mobile application.

## Download and install requisite tools:

Download and Install Docker. Installation guide for Docker is available here: <https://docs.docker.com/desktop/>

In the Command Terminal, verify that Docker is installed with the following command:

docker --version

An adequately installed docker would display the docker version and build number.

docker version 20.10.10, build b485636

Download and install NodeJs from the official NodeJS download page:

<https://nodejs.org/en/download/>

Download and Install Git. The installation guide for GIT is available here:   
<https://git-scm.com/downloads>

In the Command Terminal, verify that GIT is installed with the following command:

git --version

An adequately installed git would display the git version and build number.

git version 2.19.1.windows.1

### Install and run the Wireguard VPN Server (Recommend)

Instructions for installing, configuring, and running the Wireguard server is available here: <https://www.cyberciti.biz/faq/ubuntu-20-04-set-up-wireguard-vpn-server/>

Please verify that your Wireguard VPN Server is running by downloading the Wireguard Mobile App and configuring it to connect to the WireGuard VPN Server.   
  
Wireguard Server is not needed to run the Dossier Management System but recommended in production to secure data traffic between the API Backend and Mobile App users.

### Install and run the Database System

* 1. Run the following docker command to pull the latest MongoDB Image from Docker Hub

docker pull mongo

* 1. Start a Mongo instance using the downloaded MongoDB image with the following command.

docker run --name some-mongo -d mongo:tag

*Some-mongo* is the name you want to assign to your container, and the *tag* specifies the MongoDB version you want. By default, the authentication is not required to set up; however, it is essential to set up MongoDB admin authentication with the following docker command in a production environment.

docker run -d --network some-network --name some-mongo \

-e MONGO\_INITDB\_ROOT\_USERNAME=mongoadmin \

-e MONGO\_INITDB\_ROOT\_PASSWORD=secret mongo

### Launch the API Backend System

1. Clone and download the Dossa project file from Github

git clone git@github.com:thachp/dossa.git

1. Create a file .env.local in the project root directory and fill the following variables with your string.

APPLICATION\_ID=""

MASTER\_KEY=""

JAVASCRIPT\_KEY=""

FILES\_SUBDIRECTORY="../../appdata/incidents"

ENCRYPTIONKEY=""

MONGODB\_URL="mongodb://localhost:27017/dossa "

GRAPHQL\_SERVERURL="http://localhost:1337/graphql"

SERVER\_URL="http://localhost:1337/parse"

VPN\_IPS="::ffff:127.0.0.1"

1. Change into the /api folder and run the following commands to start the API backend.

cd /api

yarn install

yarn start:prod

The last command launches the API services at the URL <http://localhost:1337/>

### Launch the Web Application

1. From the root directory, change into the /web directory.
2. Create a file .env.local and fill the following variables with your string. Note that the APPLICATION\_ID must match the APPLICATION\_ID in the backend system from the previous instructions. Same with the JAVASCRIPT\_KEY variable.

REACT\_APP\_APPLICATION\_ID=""

REACT\_APP\_JAVASCRIPT\_KEY=""

1. Run the following commands to the start web application.

yarn install

yarn start

The last command launches a web service at the URL <http://localhost:3000>

## Launch the Mobile Application (Optional)

1. From the root directory, change into the /mobile directory
2. Edit the file /common/constants/settings.constant.ts to match those in the .env.local file from the Backend configuration.

export const HOSTNAME = "";

export const PARSE\_API = "https://example.com/parse";

export const PARSE\_GRAPHQL = "http://example.com/graphql";

export const APPLICATION\_ID = "";

export const JAVASCRIPT\_KEY = "";

1. Run the following commands to the mobile application in the Expo environment.

yarn start

For more information about EXPO, visit <https://docs.expo.dev/>

### Setup Nginx for Reverse Proxying (Production)

The following installation and configuration instructions are only necessary if Dossa is to be deployed onto a production environment. The operating system used will be Ubuntu Bionic.

1. Update the APT package cache and install the Nginx web server using the package manager:

sudo apt install

sudo apt install Nginx

1. Edit the Nginx.conf file to match the following Nginx configuration.

server {

listen 80;

listen [::]:80;

access\_log /var/log/nginx/reverse-access.log;

error\_log /var/log/nginx/reverse-error.log;

location / {

proxy\_pass http://127.0.0.1:3000;

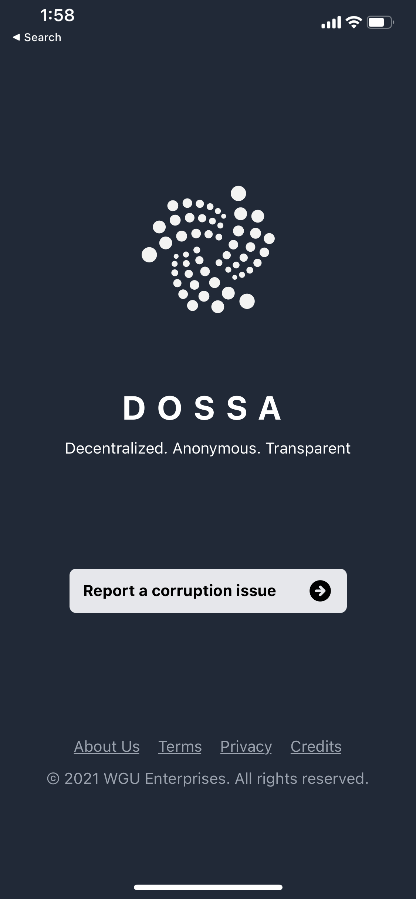
}

}

For instructions on configuration TLS on Nginx, please visit: <http://nginx.org/en/docs/http/configuring_https_servers.html>

## Submit a corruption incident through DOSSA, the Mobile Application

1. Click the “Report a corruption issue” button to begin. The goal of the application is to allow users to submit corruption issues anonymously, so user registration is not necessary. Figure 2 shows the location of the “Report a corruption issue” button located at the bottom of the screen.

*Text

Description automatically generated with low confidenceGraphical user interface, application, Teams

Description automatically generated*

Figure 2 Figure 3 Figure 4

1. Filled in all required fields (Figure 3). The form would throw form validation errors if the user forgot to fill the required fields. The Report Corruption form requires that the user…
   1. Select the “Type of Corruption.”
   2. Select up to four “Involving institutions.”
   3. Provide a short description of the incident.
   4. Optionally, the user may include hashtags
   5. Optionally, the user may include attachments of the following mime types: images, audio, video, pdf, and plain text.
2. Once all required files have been filled, press the Submit button located in the top right corner of the screen to submit the incident. The user will be taken to the Submitting Report screen (Figure 4), with various indicators showing uploading status. The user may exit the screen anytime.

## Access the Dashboard as an Analyst

1. Go to the URL <https://dossa.network/#/login>. The user will see the authentication screen. Unlike a traditional authentication screen where a user typically enters a username and password to log in, the Dossier Management System uses a public key/hash key approach to authentication. Administrators of an organization will provide this public/hash key. For demonstration purposes, the demo public key and hash has between generated:

Public Key: analyst

Key Hash: 1234

1. Enter *analyst* into the public key field and *1234* into the Key Hash field. Press the Connect button.
2. Upon successful authentication, the user will be taken to the Dashboard for a user with an analyst role, where the logged user will have the ability to view the following features:
   1. Number of institutions reported of corruptions
   2. Number of activists currently engage in fighting corruptions
   3. A list of recent institutions associated with corruptions
   4. A list of recent incidents reported by activists.
   5. An incident history chart by months.

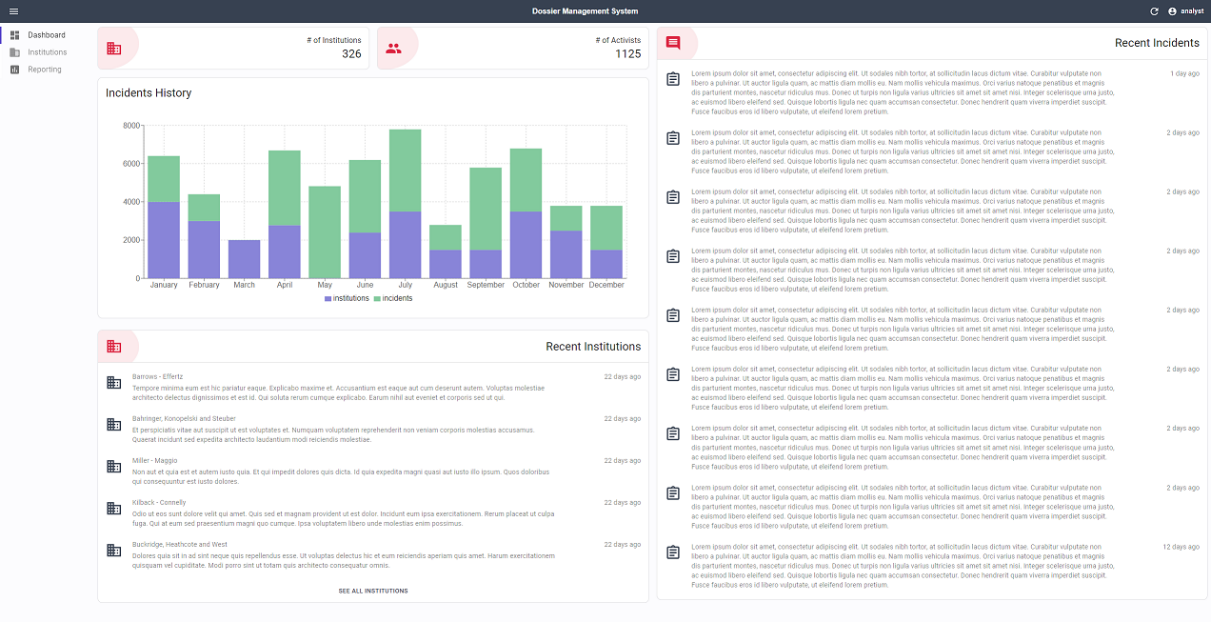


Figure 5. Dashboard view for users with Analyst role

## Access the Institutions List as an Analyst

1. Given the user is logged in with an analyst role type account, when clicked on the Institution link located on the sidebar. The user will be taken to the Institution List.
2. The user will have the ability to do the following actions with the list of institutions:
   1. Search for an institution by name.
   2. Filter institutions by corruption types
   3. Filter institutions by cases, whether there are cases created against institutions
   4. Filter by people, whether there is high profile corruption against a person working at the institutions
   5. Sort by name, description, email, and phone number
   6. Export the list of institutions
   7. Navigate between pages using the pagination tool at the footer.

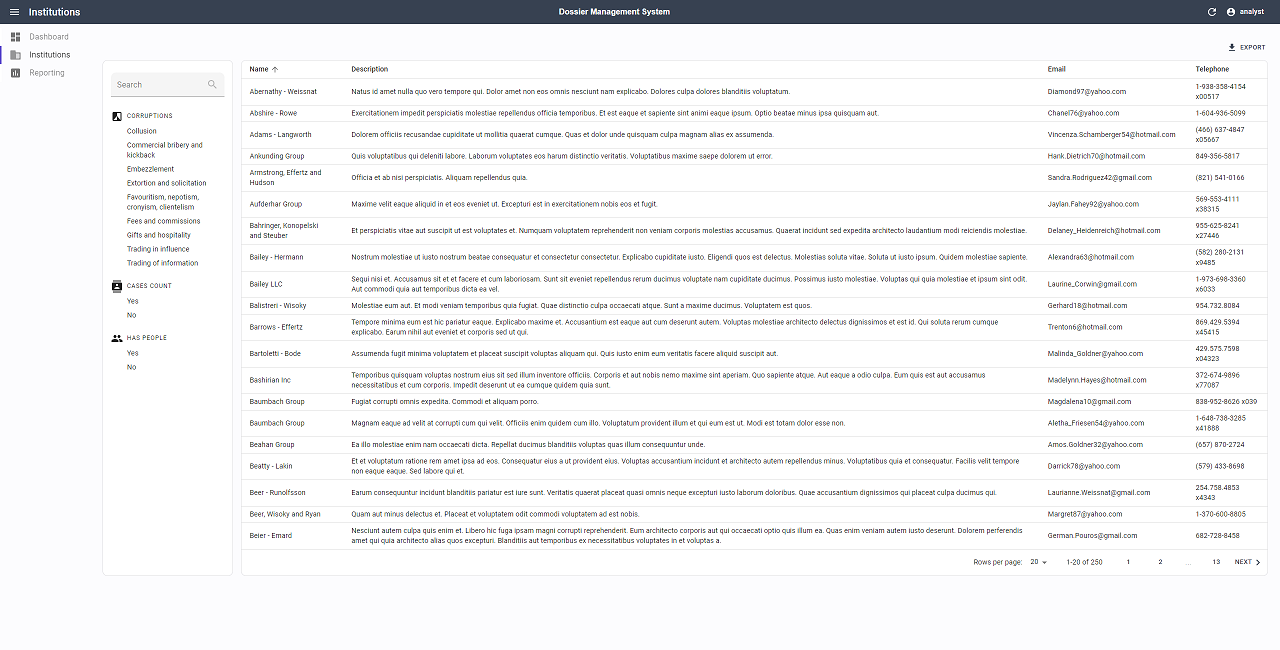


Figure 6. Institution List users with Analyst role

## Access the Reporting as An Analyst

1. Given that the user is logged in with an analyst role type account, the Reporting link is located on the sidebar. The user will be taken to the Reporting scene.
2. At the Reporting Scene, the user will be able to access the following features:
   1. A list of corruption types
   2. Corruption in Percentage as reported by incidents
   3. Trending Hashtags