

3D Model Binary Vision System

Technical Installation Manual

Flap_Jacks

Rani Arraf
Quinn du Piesanie
Jacobus Janse van Rensburg
Steven Visser
Marcus Werren

1 Introduction

The 3D Model Binary Vision Project system is a Web-based application. There are multiple packages that need to be installed and configured prior to the deployment of the web application.

The system front-end is written using HTML, CSS and Javascript. This means that to access the system when it is deployed, the user will require an official internet browser, such as Google Chrome.

The system also has a back-end. The back-end is comprised of a Javascript API and a MongoDB database. The API that was created for the system is RESTful, and uses Node/NPM to run. This means that for the system to be deployed installed correctly, the machine that runs the web-server must have both MongoDB and NodeJS installed.

Instructions and more details about the installation of these packages can be seen in the Prerequisites section of this document.

2 Prerequisites

2.1 Back End

It is recommended to rent a Virtual Private Server that offers Domain Hosting for this system.

<https://www.domains.co.za/vps-hosting>

This is an inexpensive hosting website that works very well for this systems needs. Recommended System requirements for the back-end include:

- Processor: 1 CPU Core
- Memory: 1 GB RAM
- Storage: 10 GB free
- Operating System: Ubuntu LTS 20.04
- A stable internet connection is required

In order to run the system, certain packages have to first be installed on the backend. First, Node.JS and NPM have to be installed. This can be done by typing in terminal:

```
curl -sL https://deb.nodesource.com/setup_12.x — sudo -E bash -
```

`sudo apt install nodejs`

Below is what it will look like. If you are prompted to allow the download, then enter Y and press the return key(enter) on your keyboard.

```
steven@ldap:~$ curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash -
steven@ldap:~$ sudo apt install nodejs
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libc-ares2 libhttp-parser2.7.1 nodejs-doc
The following NEW packages will be installed:
  libc-ares2 libhttp-parser2.7.1 nodejs nodejs-doc
0 upgraded, 4 newly installed, 0 to remove and 38 not upgraded.
Need to get 5,606 kB of archives.
After this operation, 24.7 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

For all storage, the API uses a MongoDB database that is created the first time the application is run. In order to install mongoDB, type the following in terminal:

`sudo apt-get install mongodb`

```
steven@ldap:~$ sudo apt-get install mongodb
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libboost-fs1.65.1 libboost-iostreams1.65.1 libboost-program-options1.65.1
  libboost-system1.65.1 libgoogle-perftools4 libpcrecpp0v5 libsnappy1v5 libstemmer0d
  libtcmalloc-minimal4 libyaml-cpp0.5v5 mongo-tools mongodb-clients mongodb-server
  mongodb-server-core
The following NEW packages will be installed:
  libboost-fs1.65.1 libboost-iostreams1.65.1 libboost-program-options1.65.1
  libboost-system1.65.1 libgoogle-perftools4 libpcrecpp0v5 libsnappy1v5 libstemmer0d
  libtcmalloc-minimal4 libyaml-cpp0.5v5 mongo-tools mongodb mongodb-clients mongodb-s
  mongodb-server-core
0 upgraded, 15 newly installed, 0 to remove and 38 not upgraded.
Need to get 53.5 MB of archives.
After this operation, 217 MB of additional disk space will be used.
Do you want to continue? [Y/n] _
```

This is not required, however if you would like to manual view and alter the database, MongoCompass is needed in order to connect to the database. Enter the following command in terminal to download it:

`sudo apt-get install mongodb-compass`

```
steven@ldap:~$ sudo apt-get install mongodb-compass
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

2.2 Front End

System requirements for the user:

- Processor: Any processor that supports SSE2
- Memory: Atleast 2GB of RAM
- Storage: 2GB Free
- Operating System: Windows 10 / Ubuntu / MacOS
- A stable internet connection is required to use the application

For accessing the web server that is created, you will require a modern and up to date browser. A selection of recommended browsers are listed below.

- Google Chrome v84.0.4147.x
- FireFox v78
- Brave v0.55

The system does not work correctly when using the Safari Browser. Issues include discolouring of certain elements.

3 Installation

If you have direct access to the terminal of the server, then you can skip this step and look at the step on navigating to the correct directory.

If you are going to access the server remotely then what you have to do, is Secure Shell into the server. To do this, you open the terminal on your system and type the following command:

```
ssh username@serverip
```

and then press enter. It should prompt you for the password as shown below.

```
C:\Users\steve>ssh root@41.7 .1 9.148
root@41.7 .1 9.148's password:
```

Enter your password and if it is correct, then you will see a screen that resembles the screen below.

```
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-29-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Thu 20 Aug 2020 01:51:11 PM SAST

System load:          0.0
Usage of /:           12.2% of 22.58GB
Memory usage:         21%
Swap usage:           0%
Processes:            90
Users logged in:      0
```

Once you have secure shelled into the system, or have logged in directly (for the users who have direct access to the server machine), you will have to create a new directory, this will be where you install the repository. Type the following into the terminal to make a directory. We called ours /test.

```
sudo mkdir /test
```

and press enter. The directory should have been made and you can now transfer files to it.

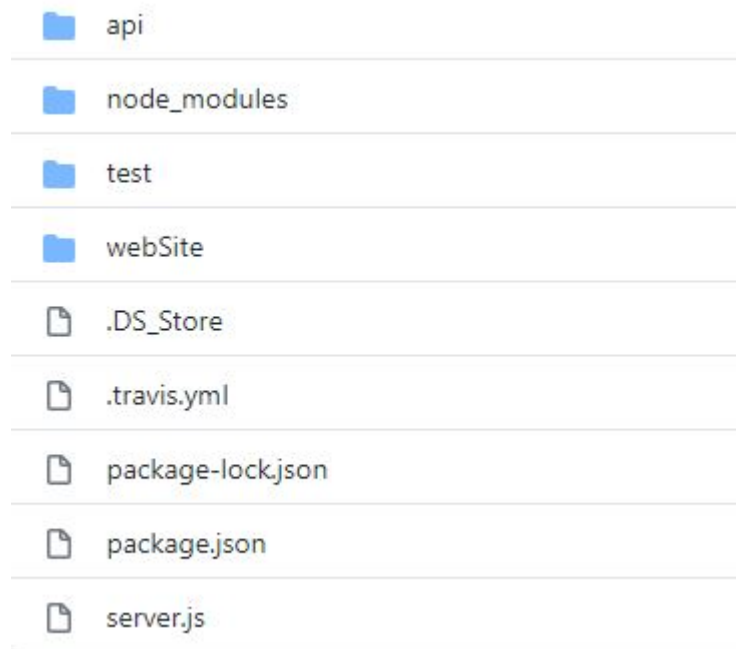
```
root@flapjacks:~# sudo mkdir /test1
root@flapjacks:~#
```

Now you go back to your own machine. You need to download the master branch off of our github repository. The repository can be downloaded from:

<https://github.com/COS301-SE-2020/3D-model-Binary-Vision>

When the zip file has completed its download, unzip it and navigate to the

Application directory. This is where the system is sitting.



You will have to send the files to the server now. The easiest way to do this is by using the terminal scp command. Navigate your machine's terminal to the Application/ directory of the downloaded files and type the following:

```
scp -r * username@serverip:/directoryOnServer
```

- scp is a secure copy command
- -r says that you want to include directories
- the asterisk says you want to move everything

```
Microsoft Windows [Version 10.0.18362.1016]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\steve>scp -r * root@41.71.19.148:/test
```

Now that the files are on the server, you will want to secure shell back into it and navigate to the directory that you installed the repository. In the example below, the repository was stored in the directory /test.

To change to this directory, simply type the following in terminal:

```
cd /test
```

and press enter. Type `ls` in terminal to see what files are in the directory to confirm you are in the right place.

```
root@flapjacks:~# cd /test
root@flapjacks:/test# ls
api  JacoCookie  node_modules  package.json  package-lock.json  server.js  test  webSite
root@flapjacks:/test#
```

If you correctly installed the repository, you will see a file called `server.js`. The next step for installation is to go into this file and change line 4 to have the port that you want the server to run over. For a normal webserver, we will want to use port 80. However, if you have an SSL, you will want to use port 443.

In terminal, type `'vim server.js'` and press enter. Press the down arrow until you are on line 4. Press the a-key to edit the line, use the right arrow to navigate to the number and change it to 80.

Now that the file has been edited. Press Esc key to exit the editor and type `':wq'` and press enter again. This will save the updates to the `server.js` file.

```
//variables needed
var express = require('express'),
    app = express(),
    port = process.env.PORT || 80,
    mongoose = require('mongoose'),
    bodyParser = require('body-parser'),
    serveStatic = require('serve-static');
```

The system is now ready to be run! The next section of this document will detail the instructions on how to run and access the system.

4 Security

As an extra measure for security, and as a requirement from most browsers to access some of the content of the application, you will need to get an SSL certificate and install it into the configuration file for the server.

[Click here](#) to see an extensive guide on how to get an SSL Certificate

[Click here](#) for an extensive guide on installing the SSL to the configuration

5 Deployment/Running

Once you have installed all of the prerequisites and correctly unpacked the repository, you will have to run the server to be able use the system.

In terminal, you will have to navigate to the directory that you unpacked the repository. Once there, type the following command:

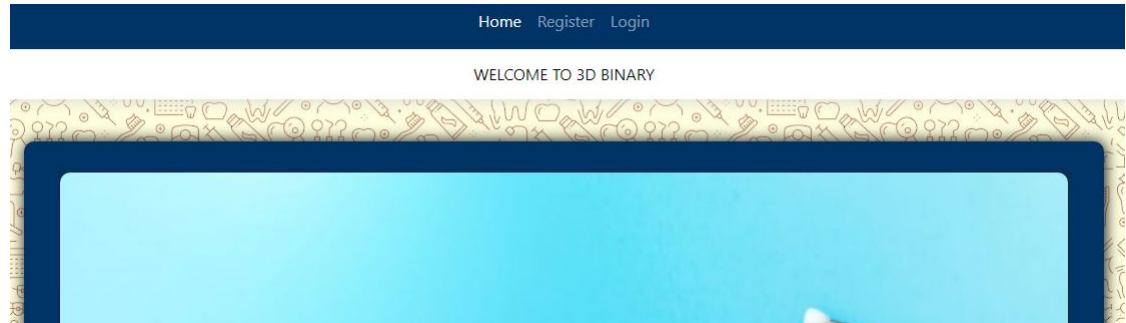
```
node server.js
```

and press enter. If all of the prerequisites have been installed correctly then you will see the following screen:

```
root@flapjacks:/test# node server.js
Restful API for 3DModel Flap Jacks Started On Port 80
```

The server is now running on the port that you specified.

Now on your own device, you can open your browser and go to the web address that your VPS is set up on. In our case, we go to flapjacks.goodx.co.za. Below is the page that you should see:



For further information about the system, and a detail manual on how to use it, please follow the link below to download the user manual:

[Click here to be redirected to the User Manual!](#)