



BushBuddy

# Technical Installation Guide

*Version 1*

## Team members

Ruan Esterhuizen (u23532387)

Ruben Hannes Gadd (u23633353)

Raphael Rato (u22887581)

Tom Schulz (u05039364)

Jean Steyn (u22537229)

## Contact

[g24capstone@gmail.com](mailto:g24capstone@gmail.com)

# Table of Contents

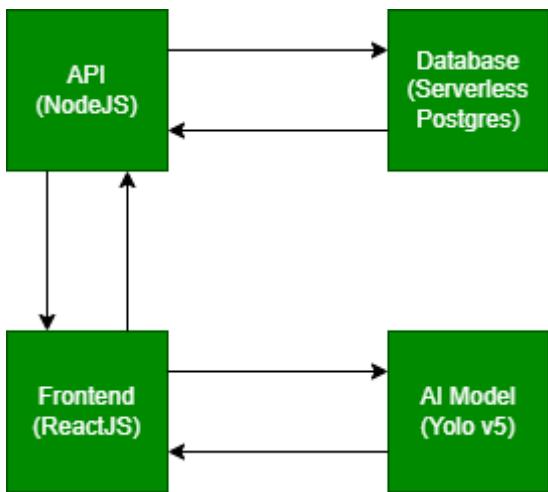
---

Introduction .....	3
System Overview.....	3
Prerequisites .....	3
Hardware requirements .....	3
Software requirements.....	3
Major Software Packages.....	4
Prerequisite Software Installation.....	5
Windows Installation.....	5
Linux Installation.....	6
MacOS Installation .....	7
Running and Deployment.....	8
Further Reading .....	12

# Introduction

This guide details all necessary steps and information needed to configure a user's system to install and run the BushBuddy system. Follow these steps carefully to ensure successful installation.

## System Overview



**Please note** that this is a high-level diagram that is only used to illustrate what the subsystems are and how they interact, it is subject to change and a more in-depth explanation of the system architecture can be found in the [SRS Document](#).

# Prerequisites

## Hardware requirements

	Minimum	Recommended
<b>CPU</b>	2 GHz, Dual core	2.5 GHz, Quad core
<b>Memory</b>	4 GB	8 GB
<b>Other</b>	Webcam and microphone	

## Software requirements

Software	Version	Download
<b>Operating System</b>	Windows 10/11 MacOS 15 Sequoia Linux (64-bit)	
<b>Web Browser</b>	Google Chrome (Latest) Safari ~18.5 (Latest) Firefox (Latest)	<a href="#">Chrome Download</a> <a href="#">Built in (MacOS only)</a> <a href="#">Download Firefox</a>
<b>Git</b>	2.50.1	<a href="#">git-scm.com/downloads</a>



<b>NodeJS</b>	20.13.1	<a href="https://nodejs.org/en/download/">nodejs.org/en/download/</a>
<b>npm (Node Package Manager)</b>	10.5.2	Included with NodeJS
<b>Python</b>	3.12.10	<a href="https://python.org/downloads/">python.org/downloads/</a>
<b>pip</b>	25.0.1	Included with Python

Additionally, a text editor of your choice should also be used (**VS Code** is recommended).

## Major Software Packages

Package	Purpose	Version
<b>ReactJS*</b>	User Interface	18.2.0
<b>Axios*</b>	HTTP Requests	1.11.0
<b>Jest*</b>	Unit Testing	29.6.2
<b>Bootstrap*</b>	User Interface Styling	5.3.7
<b>Leaflet*</b>	Maps	1.9.4
<b>Dotenv*</b>	Environment Variable Management	17.2.1
<b>Bcrypt*</b>	Password encryption	6.0.0
<b>Yolo v5</b>	AI Image Recognition	8.3.179

\* Installation of Javascript packages is automatically handled by npm.

# Prerequisite Software Installation

After completing these steps, you can commence to [Running and Deployment](#)

## Windows Installation

### Step 1: Install Git

Go to [git-scm.com/downloads/win](https://git-scm.com/downloads/win) and follow the instructions to install Git.

**Download for Windows**

[Click here to download](#) the latest (2.50.1) x64 version of **Git for Windows**. This is the most recent [maintained build](#). It was released **40 days ago**, on 2025-07-08.

**Other Git for Windows downloads**

[Standalone Installer](#)  
[Git for Windows/x64 Setup](#).  
[Git for Windows/ARM64 Setup](#).  
[Portable \("thumbdrive edition"\)](#)  
[Git for Windows/x64 Portable](#).  
[Git for Windows/ARM64 Portable](#).

**Using winget tool**  
Install [winget tool](#) if you don't already have it, then type this command in command prompt or Powershell.  
`winget install --id Git.Git -e --source winget`

The current source code release is version 2.50.1. If you want the newer version, you can build it from the [source code](#).

**Optional:** A Git GUI Client (such as [GitHub Desktop](#)) can additionally be used, but following instructions will use the Git Command Line Tool.

### Step 2: Install NodeJS

Go to [nodejs.org/en/download](https://nodejs.org/en/download) and follow the instructions to install NodeJS (Using the Windows Installer is recommended).

**Download Node.js®**

Get Node.js® v22.18.0 (LTS) for [Windows](#) using [Docker](#) with [npm](#)

**Info** Want new features sooner? Get the [latest Node.js version](#) instead and try the latest improvements!

```
1 # Docker has specific installation instructions for each operating system.
2 # Please refer to the official documentation at https://docker.com/get-started/
3
4 # Pull the Node.js Docker image:
5 docker pull node:22-alpine
6
7 # Create a Node.js container and start a Shell session:
8 docker run -it --rm --entrypoint sh node:22-alpine
9
10 # Verify the Node.js version:
11 node -v # Should print "v22.18.0".
12
13 # Verify npm version:
14 npm -v # Should print "10.9.3".
```

PowerShell [Copy to clipboard](#)

Docker is a containerization platform. If you encounter any issues please visit [Docker's website](#)

Or get a prebuilt Node.js® for [Windows](#) running a [x64](#) architecture.

[Windows Installer \(.msi\)](#) [Standalone Binary \(.zip\)](#)

## **Step 3: Install Python**

Go to <https://www.python.org/downloads/> and follow the instructions to install the latest version of Python (pip is included in this installation).



## **Linux Installation**

### **Step 1: Install Git**

Open the Terminal and run the following commands to install Git.

```
sudo apt update  
sudo apt install git
```

### **Step 2: Install NodeJS and npm**

```
sudo apt install node  
sudo apt install npm
```

### **Step 3: Install Python and pip**

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



# MacOS Installation

## Step 1: Install Homebrew (Optional)

Open the Terminal and run the following commands:

```
$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

## Step 2: Install Git

The following steps assume that Homebrew is used, for alternative methods go to [here](#).

```
$ brew install git
```

## Step 3: Install NodeJS

Go to <https://nodejs.org/en/download> and follow the instructions (Downloading the .pkg installer is recommended).

The screenshot shows the official Node.js download page. It features a search bar at the top with dropdown menus for selecting the version (v22.18.0 LTS), platform (macOS), and method (Docker). Below the search bar is a blue header bar with the text "Info Want new features sooner? Get the [latest Node.js version](#) instead and try the latest improvements!". The main content area contains a code block with 14 numbered lines of Bash script. Lines 1-3 are comments about Docker. Line 4 starts a Docker pull command. Lines 5-9 show a Docker run command to start a Node.js container. Lines 10-12 verify the Node.js and npm versions. Lines 13-14 verify the npm version again. At the bottom of this code block is a "Bash" label and a "Copy to clipboard" button. Below the code block is a note: "Docker is a containerization platform. If you encounter any issues please visit [Docker's website](#)". Further down, there's another section for prebuilt installers, with buttons for "macOS Installer (.pkg)" and "Standalone Binary (.gz)".

## Step 3: Install Python and pip

Pip should automatically be installed along with Python using the following command:

```
$ brew install python
```

# Running and Deployment

**Note:** All the following terminal commands use the Windows Commands Prompt syntax, which may vary for Linux and MacOS users.

## Step 1: Clone the GitHub Repo

Open the terminal in your file location of choice and run the following command:

```
git clone https://github.com/COS301-SE-2025/AI-Powered-African-Wildlife-Detection
```

After the repo has successfully been cloned, a new directory should be created (“AI-Powered-African-Wildlife-Detection”), cd into that folder and/or open it in your text editor of choice.

## Step 2: Setting up environment variables

### 2.1 Frontend environment variables

Create a file named “`.env`” in the “`./frontend`” directory. This file should have the following contents:

Variable Name	Description	Variable Value
<code>REACT_APP_URL</code>	The URL of the API used	<code>http://localhost:3000</code>
<code>PORT</code>	The default port for the React app	3001

Variable values can be configured to suit your environment.

### 2.2 API environment variables

Create a file named “`.env`” in the “`./api`” directory. This file should have the following contents:

Variable Name	Description	Variable Value
<code>ENVIRONMENT</code>	Used for deployment on Render	dev or prod
<code>DATABASE_URL_ENV</code>	The URL of the Postgres DB	Connection string of your DB
<code>JWT_SECRET</code>	Used for user authentication	Any string of your choice
<code>ALLOWED_ORIGINS</code>	Used for CORS	<code>http://localhost:3000</code>

**Note:** Entries in a `.env` file have the following format:

```
<variable_name> = <variable_value>
```

For example:

```
ALLOWED_ORIGINS = http://localhost:3001
```

## Step 3: Running the api

1. In the terminal, cd into the **./frontend** directory.
2. Run the following commands to install dependencies and run the api.

```
npm install  
npm run
```

When the api starts running, it will print the following output to the terminal

```
Restarting 'src/server.js'  
  
[dotenv@17.2.0] injecting env (4) from .env (tip: ⚡ write to custom object with {  
processEnv: myObject })  
  
[dotenv@17.2.0] injecting env (0) from .env (tip: ⚡ enable debug logging with {  
debug: true })  
  
[dotenv@17.2.0] injecting env (0) from .env (tip: ⚡ prevent building .env in  
docker: https://dotenvx.com/prebuild)  
  
Auth service is running on port 4001  
Discovery service is running on port 4002  
Sightings service is running on port 4003  
Post service is running on port 4004  
Server is running on port 3000  
Database connected successfully
```

## Step 4: Running the AI model

1. Install the latest version of the Yolo v5 package by running the following command:

```
pip install yolo
```

2. To run the pre-trained model, naviagte to the **AI/python** directory and run the following command.

```
python .\test_yolo.py
```

Alternatively, you can open the file in your text editor and a “run” button should appear, depending on the text editor and extensions installed.

In the “test\_yolo.py” file an image is loaded as the model’s input (on line 9 to be exact), you can changes the default image entered from there.

3. Training data is fetched from the iNatrualist database in “**AI/python/fetch\_images.py**”
4. Images are annotated and prepared for training in “**AI/python/filter\_annotations.py**” and “**AI/python/fix\_annotations.py**”.

## Step 5: Running the frontend

1. In the terminal, cd into the `./api` directory and run the following commands:

```
npm install  
npm start
```

On successful execution, the default browser should open and load the application. The following console output should be printed:

```
> bush-buddy@0.2.0 start  
> react-scripts start  
...  
Starting the development server...  
Compiled successfully!  
You can now view bush-buddy in the browser.  
Local:          http://localhost:3001  
On Your Network: http://192.168.56.1:3001  
Note that the development build is not optimized.  
To create a production build, use npm run build.  
webpack compiled successfully
```

### Viewing the frontend on mobile devices

Although BushBuddy works fine on desktop browsers, it is intended for mobile devices. To access the locally running version of the app, the following steps can be followed:

1. Any mobile device with the newest version of Chrome, Safari or FireFox can be used.
2. Make sure your desktop and mobile device are connected to the same network.
3. While the app is running on your desktop, open the mobile browser.
4. In the console output, an address with your desktop's IP address and port will appear. In the above example, the address is:

```
http://192.168.56.1:3001
```

5. Enter the address in your mobile browser and BushBuddy should open, where it can be installed as a PWA.

**Note:** Viewing BushBuddy on a mobile device via your network is highly dependent on your network and device setup and may not work, thus it is not recommended. More info for troubleshooting can be found [here](#) and [here](#) (You will need it).

## Further Reading

---

Now that BushBuddy is set up and running on your system, please see the [User Manual](#) to learn how to use and navigate the system.