Municipality Permit Chatbot System (MPCS)

Programmer's Guide

University of Maryland Global Campus (UMGC) SWEN 670

Software Version: #2

Document Version: 1.1

August 4, 2020

**Prepared by UMGC ChatBot Team**:

Rusty Baker

Sepribo Taylor-Harry

Matthew Slaymaker

Nathaniel Muesing

Subhash Gandhi Vallala

Joshua Piersol

Programmer Guide Approvals

|  |  |  |
| --- | --- | --- |
| Name | Signature | Date |
| Approved by:  "Software Lead" |  |  |
| Approved by:  "Project Manager" |  |  |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Date | Name | Description of Change(s) |
| 1.0 | July 23 2020 | Vallala, Subhash Gandhi | Initial Release |
| 1.1 | August 4, 2020 | Sepribo Taylor-Harry, Nathaniel Muesing | Incorporated Stakeholder feedback |

Contents

[1. Introduction 7](#_Toc47472622)

[1.1 Purpose 7](#_Toc47472623)

[1.2 Intended audience 7](#_Toc47472624)

[1.3 Technical project stakeholders 7](#_Toc47472625)

[1.4 Definitions, acronyms and abbreviations 8](#_Toc47472626)

[2. System Architecture 8](#_Toc47472627)

[2.1 Architectural Design 8](#_Toc47472628)

[2.2 Architectural Overview 8](#_Toc47472629)

[3. Setting up the application 9](#_Toc47472630)

[3.1 Downloading Git Bash 9](#_Toc47472631)

[3.2 Downloading the latest code 17](#_Toc47472632)

[3.3 Installing the frontend 17](#_Toc47472633)

[3.4 Installing the backend 23](#_Toc47472634)

[3.5 Running the frontend 24](#_Toc47472635)

[3.6 Running the backend 25](#_Toc47472636)

[3.7 Running unit tests 25](#_Toc47472637)

[3.8 Running end-to-end tests 26](#_Toc47472638)

[4. Code structure 26](#_Toc47472639)

[4.1 Frontend 26](#_Toc47472640)

[4.1.1 Source code 26](#_Toc47472641)

[4.1.1.1 Account 28](#_Toc47472642)

[4.1.1.2 Home 28](#_Toc47472643)

[4.1.1.3 Users 28](#_Toc47472644)

[4.1.1.4 Zone-dashboard 28](#_Toc47472645)

[4.1.2 Assets 28](#_Toc47472646)

[4.1.3 Environments 29](#_Toc47472647)

[5. Code scaffolding 29](#_Toc47472648)

[5.1 Frontend 29](#_Toc47472649)

**Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.**

**List of Tables**

[Table 1 8](#_Toc46523048)

[Table 2 9](#_Toc46523049)

**List of Figures**

[Figure 1 Git Bash Install Website 11](#_Toc47508006)

[Figure 2 Git Bash Installer SaveFile 11](#_Toc47508007)

[Figure 3 Git Bash Installer Setup 12](#_Toc47508008)

[Figure 4 GitBash Select Destination 12](#_Toc47508009)

[Figure 5 GitBash Select Components 13](#_Toc47508010)

[Figure 6 Select Start Menu Folder 13](#_Toc47508011)

[Figure 7 Choose the default editor for GIT 14](#_Toc47508012)

[Figure 8 Selec Use Nano editor 14](#_Toc47508013)

[Figure 9. Adjust your Path environment 15](#_Toc47508014)

[Figure 10 Use HTTPS Transport Backend 15](#_Toc47508015)

[Figure 11 GitBash Finished Installation 16](#_Toc47508016)

[Figure 12 Accessing Git Bash 17](#_Toc47508017)

[Figure 13 Terminal Window Git Bash 17](#_Toc47508018)

[Figure 14 Node.js Website. Click to download the latest version. (Screen Capture of Node.js Setup) 19](#_Toc47508019)

[Figure 15 Launch the setup file. (Screen Capture of Node.js Setup) 19](#_Toc47508020)

[Figure 16 Accept the license agreement. (Screen Capture of Node.js Setup) 20](#_Toc47508021)

[Figure 17 Select the installation path. (Screen Capture of Node.js Setup) 20](#_Toc47508022)

[Figure 18 Select the installation structure. (Screen Capture of Node.js Setup) 21](#_Toc47508023)

[Figure 19 Checkmark the box to automatically install necessary tools 21](#_Toc47508024)

[Figure 20 Begin installation. (Screen Capture of Node.js Setup) 22](#_Toc47508025)

[Figure 21 Finish installation. (Screen Capture of Node.js Setup) 22](#_Toc47508026)

[Figure 22 Install necessary tools by pressing any key (recommended). 23](#_Toc47508027)

[Figure 23 How to download MongoDB Community Edition 24](#_Toc47508028)

[Figure 24 How to Start MongoDB Service 26](#_Toc47508029)

[Figure 25 umgc.city\_chatbot GitHub repository 27](#_Toc47508030)

[Figure 26 city-official-client folder 28](#_Toc47508031)

[Figure 27 List of components in city-official-client folder 28](#_Toc47508032)

**Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.**

# 1. Introduction

## 1.1 Purpose

The purpose of the Programmer's Guide is to guide current and future programmers with an understanding of how the Municipality Permit Chatbot (MPC) will application development environment is. Furthermore, this Programmer's Guide will discuss software architecture, explicitly how the developers incorporate the architecture in a development environment.

## 1.2 Intended audience

The Programmer's Guide's audience is specific to the software developers, programmers, and testers of the Municipal City Chatbot System. Furthermore, this guide will cover both the front end, client-side of the application, the back end, the server-side of the application, and testing capabilities.

## 1.3 Technical project stakeholders

Table 1 outlines the stakeholders of the MPC System. Stakeholders specific to this document are that of the Test Lead, Systems Lead, and Developer Lead.

Table 1

Technical project stakeholders

|  |  |  |
| --- | --- | --- |
| Name | E-mail address | Role |
| Professor Assadullah | mir.assadullah@faculty.umgc.edu | Stakeholder |
| Rusty Baker | rusty.j.b2016@gmail.com | Project Manager |
| Nathaniel Muesing | nkmuesing@gmail.com | Lead Developer |
| Sepribo Taylor-Harry | seprtayl@gmail.com | Lead DevOps |
| Matthew Slaymaker | slaymakerm@wmpenn.edu | Lead Testing |
| Subhash Gandhi Vallala | sgvallala@gmail.com | Lead Quality Assurance |
| Joshua Piersol | joshpwork@yahoo.com | Lead Systems |

## 1.4 Definitions, acronyms and abbreviations

Table 2

Definitions, acronyms, and abbreviations

|  |  |
| --- | --- |
| Term | Definition |
| Angular CLI | Angular Command Line Interface |
| CSS | Cascading Style Sheets |
| HTML | Hypertext Markup Language |
| IDE | Integrated Development Environment |
| JSON | JavaScript Object Notation |
| npm | Node Package Manager |
| UAT | User Acceptance Testing |
| UI | User Interface |
| UMGC | University of Maryland Global Campus |
| URL | Uniform Resource Locator |
| VS Code | Visual Studio Code |

# 

# 2. System Architecture

## 2.1 Architectural Design

The system is made up of three main components. The city official client (frontend) is an [Angular](https://angular.io/guide/setup-local) service that is the UI to manipulate data persistently stored by a Java-based API (backend). An instance of [MongoDB](https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows/) is run locally for development purposes and allows future users of the system to migrate later if they wish to use their own solution. Installation of Angular CLI (frontend) and the MongoDB (backend) database is provided in section 3.3 and 3.4 of this documentation. The chatbot component is hosted within IBM's could environment. Java API is utilized to update and manage the dialog of the chatbot.

## 2.2 Architectural Overview

The architecture of the MPC is made of three main components consists of the following:

• City Official Client Front-End: It is an Angular-based component that allows city officials to tender data to populate chatbot branches and dialog responses. Furthermore, the city hosts related city permits and regulation web pages are linked to zones. Zones can be created and tracked with geolocations utilizing OpenLayers. The OpenLayers API is utilized for both creating zone vectors and for retrieving geolocation data uploaded to OpenLayers.

• City Official Client Back-End: This element is made up of a Java handler that forms data submitted by the Front-End client for both persistent storage and a running instance of Watson Assistant. Data submitted will be lumped together based on zone and permit type. This will help with database normalization and reduce the amount of information needed to be submitted by city officials for a functional chatbot.

• Chatbot: Is provided by a cloud instance of Watson Assistant. Functionality is determined based on intents, a dialog topic importance level, and entities. Intents are groupings of responses that are created based on defined categories. Entities are words/terms that prime Watson Assistant with context to respond with.

# 3. Setting up the application

## 3.1 Downloading Git Bash

Before downloading the current software repository, one will have to ensure they have GIT in order to unpackage the repository (e.g., Checkout). If one does not have GIT, then follow the below process in setting up your development environment with GIT. To download the latest version of Git, click the following link:  [https://Git-scm.com/](https://git-scm.com/). Upon opening the website, click on the "Download for Windows" button located within the display monitor, as shown in Figure 1. Once clicked, the website will transition to a download page that prompts that GIT installer is downloading.



Figure Git Bash Install Website. Reprinted from How To Install Git Bash On Windows by Stanley Ulili. (n.d.). Retrieved from [https://www.stanleyulili.com/Git/how-to-install-Git-bash-on-windows/](https://www.stanleyulili.com/git/how-to-install-git-bash-on-windows/). Copyright n.d. By Stanley Ulili.

Upon download, the Git Bash installer for your current Windows Setup (e.g., 32-bit or 64-bit), click and being the Install Git Bash process by selecting the save file and as shown in Figure 2.

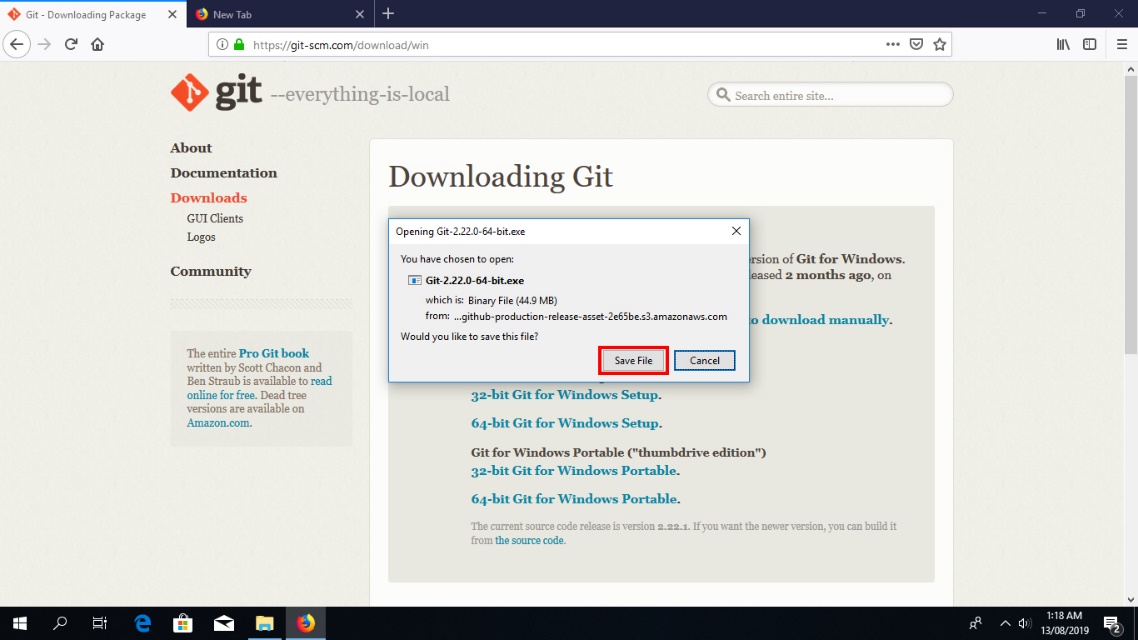


Figure Git Bash Installer SaveFile. Reprinted from How To Install Git Bash On Windows by Stanley Ulili. (n.d.). Retrieved from https://www.stanleyulili.com/Git/how-to-install-Git-bash-on-windows/. Copyright n.d. By Stanley Ulili.

Once saved, open the installer and follow the outlined standard process, as shown in Figures 3-6.



Figure Git Bash Installer Setup. Reprinted from How To Install Git Bash On Windows by Stanley Ulili. (n.d.). Retrieved from https://www.stanleyulili.com/Git/how-to-install-Git-bash-on-windows/. Copyright n.d. by Stanley Ulili

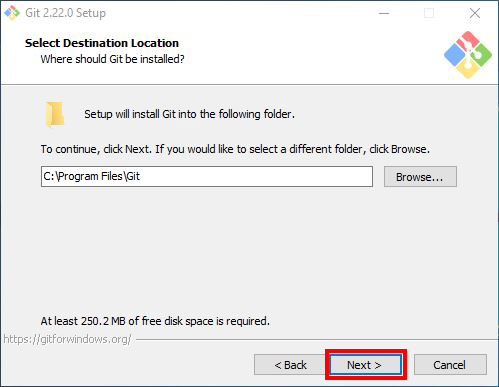
. 

Figure GitBash Select Destination. Reprinted from How To Install Git Bash On Windows by Stanley Ulili. (n.d.). Retrieved from https://www.stanleyulili.com/Git/how-to-install-Git-bash-on-windows/. Copyright n.d. by Stanley Ulili.

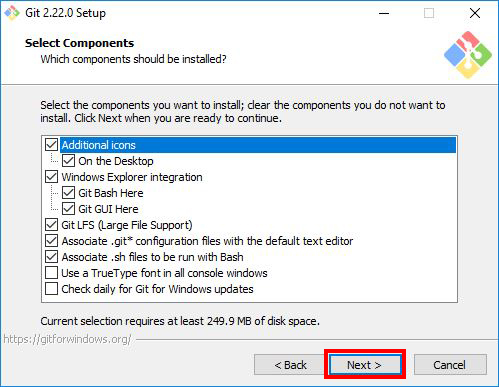


Figure GitBash Select Components. Reprinted from How To Install Git Bash On Windows by Stanley Ulili. (n.d.). Retrieved from https://www.stanleyulili.com/Git/how-to-install-Git-bash-on-windows/. Copyright n.d. By Stanley Ulili.

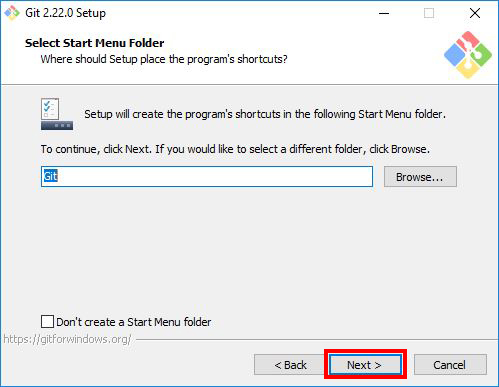


Figure Select Start Menu Folder. Reprinted from How To Install Git Bash On Windows by Stanley Ulili. (n.d.). Retrieved from https://www.stanleyulili.com/Git/how-to-install-Git-bash-on-windows/. Copyright n.d. By Stanley Ulili.

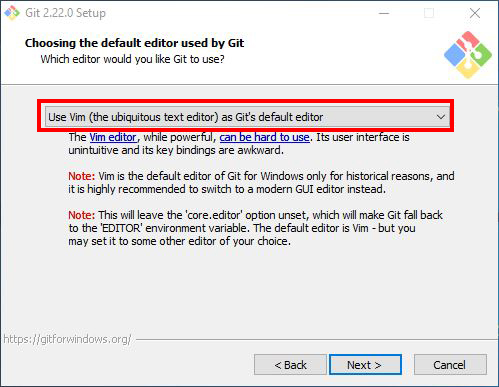


Figure Choose the default editor for GIT. Reprinted from How To Install Git Bash On Windows by Stanley Ulili. (n.d.). Retrieved from https://www.stanleyulili.com/Git/how-to-install-Git-bash-on-windows/. Copyright n.d. By Stanley Ulili.

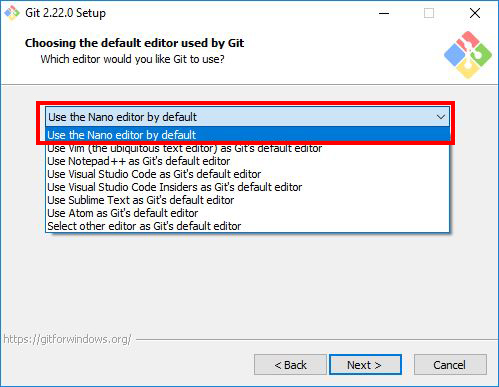


Figure Selec Use Nano editor. Reprinted from How To Install Git Bash On Windows by Stanley Ulili. (n.d.). Retrieved from https://www.stanleyulili.com/Git/how-to-install-Git-bash-on-windows/. Copyright n.d. By Stanley Ulili.

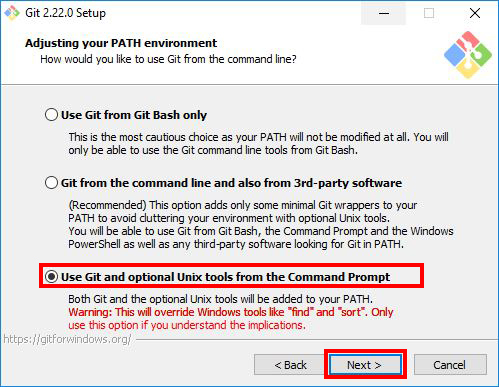


Figure . Adjust your Path environment. Reprinted from How To Install Git Bash On Windows by Stanley Ulili. (n.d.). Retrieved from https://www.stanleyulili.com/Git/how-to-install-Git-bash-on-windows/. Copyright n.d. By Stanley Ulili.

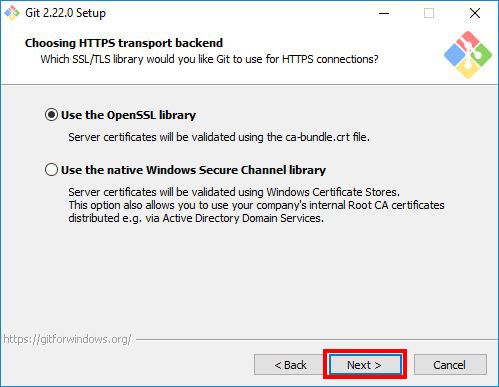


Figure Use HTTPS Transport Backend. Reprinted from How To Install Git Bash On Windows by Stanley Ulili. (n.d.). Retrieved from https://www.stanleyulili.com/Git/how-to-install-Git-bash-on-windows/. Copyright n.d. By Stanley Ulili.

For the remaining screens of the installation, click the "NEXT" button. You will complete the Git Bash installation process once you click the "Finish" button outlined in Figure 11.

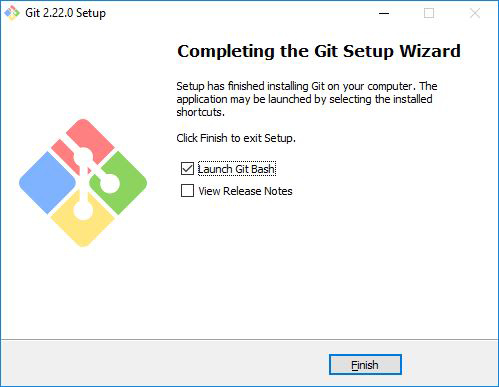


Figure GitBash Finished Installation. Reprinted from How To Install Git Bash On Windows by Stanley Ulili. (n.d.). Retrieved from https://www.stanleyulili.com/Git/how-to-install-Git-bash-on-windows/. Copyright n.d. By Stanley Ulili.

To access the Git Bash environment, simply right click on your desktop and scroll to and select "Git Bash Here" as shown in Figure 12. Upon selection, the Git Bash terminal window will appear, as shown in Figure 13. If all appears, you will have successfully installed Git Bash.

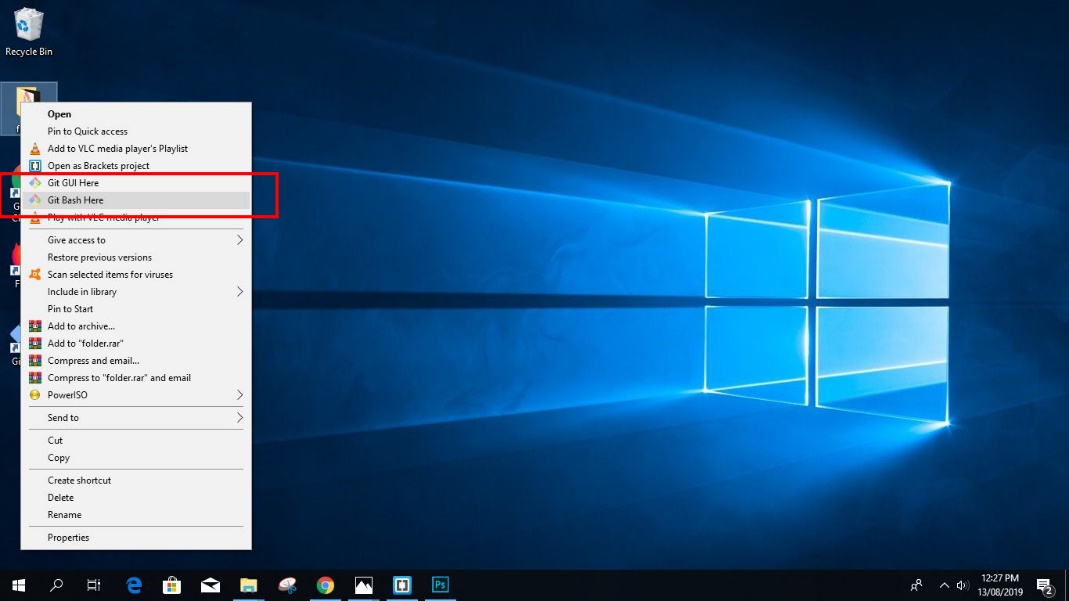


Figure Accessing Git Bash. Reprinted from How To Install Git Bash On Windows by Stanley Ulili. (n.d.). Retrieved from https://www.stanleyulili.com/Git/how-to-install-Git-bash-on-windows/. Copyright n.d. By Stanley Ulili.

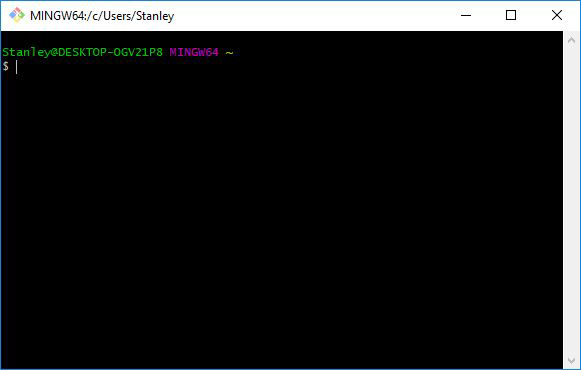


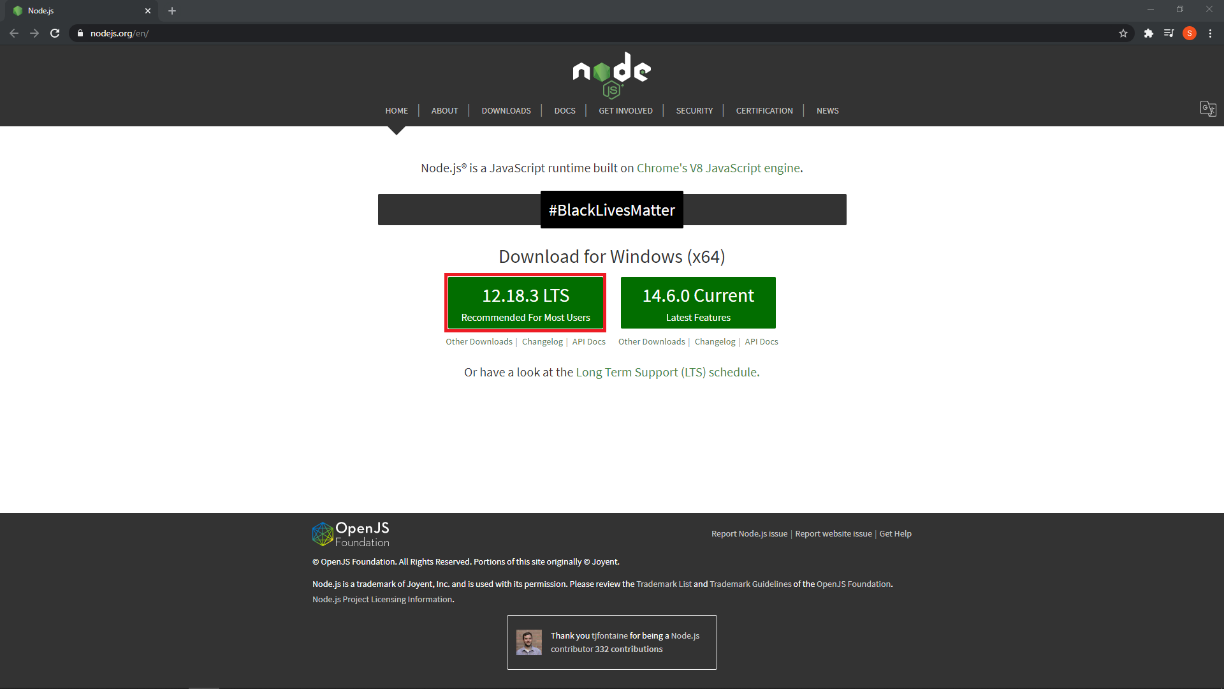
Figure Terminal Window Git Bash. Reprinted from How To Install Git Bash On Windows by Stanley Ulili. (n.d.). Retrieved from https://www.stanleyulili.com/Git/how-to-install-Git-bash-on-windows/. Copyright n.d. By Stanley Ulili.

## 3.2 Downloading the latest code

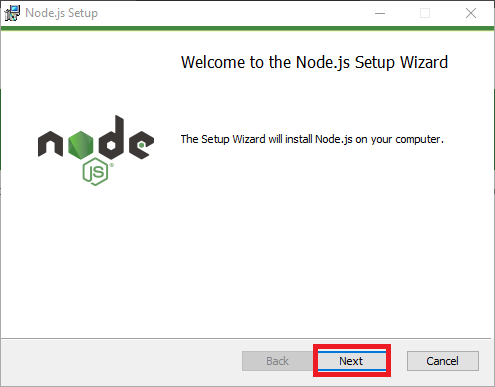
The following link outlines the current Github repository to where the current code for the MPC System can be found: [https://Github.com/umgc/umgc.city\_chatbot](https://github.com/umgc/umgc.city_chatbot). Before downloading the software, ensure that you are signed into GitHub. If you currently do not have a GitHub account, then please create and sign up before continuing. Furthermore, in order to access the MPC repository, please ensure that your GitHub account is linked to your University of Maryland Global Campus (UMGC) account. Once signed in, click on the green download button labeled "Code" button containing an arrow pointing down on the top right of the page. Select the clone with HTTPS and copy the URL into the clipboard. Open a terminal window and navigate to the folder location where you would like to download the latest code. Once at the desired location, enter the command . You now have the latest code in your system locally.

## 3.3 Installing the frontend

To install the frontend, navigate to the folder within the cloned repository folder. Download and install the latest Node.js from <https://nodejs.org/en/>. Follow the following steps to install Node.js:



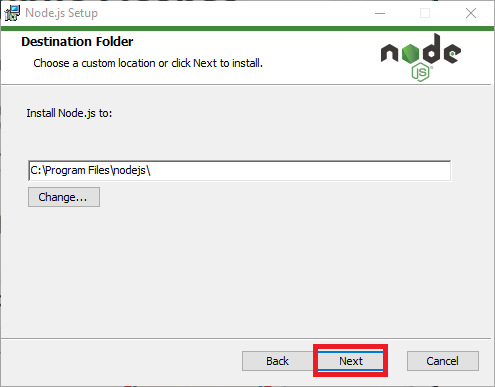
*Figure 14 Node.js Website. Click to download the latest version. (Screen Capture of Node.js Setup)*



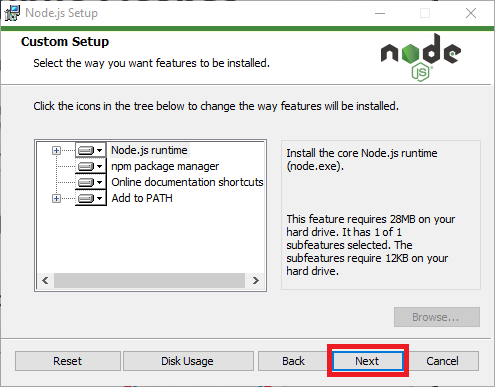
*Figure 15 Launch the setup file. (Screen Capture of Node.js Setup)*



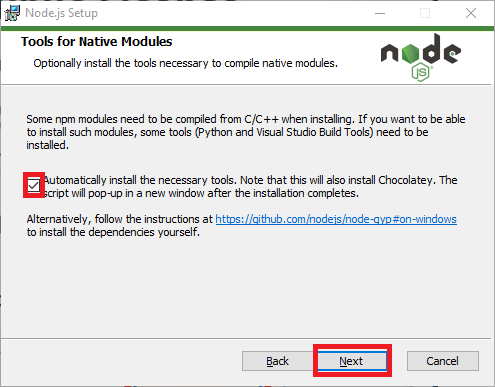
*Figure 16 Accept the license agreement. (Screen Capture of Node.js Setup)*



*Figure 17 Select the installation path. (Screen Capture of Node.js Setup)*



*Figure 18 Select the installation structure. (Screen Capture of Node.js Setup)*



*Figure 19 Checkmark the box to automatically install necessary tools (recommended). (Screen Capture of Node.js Setup)*

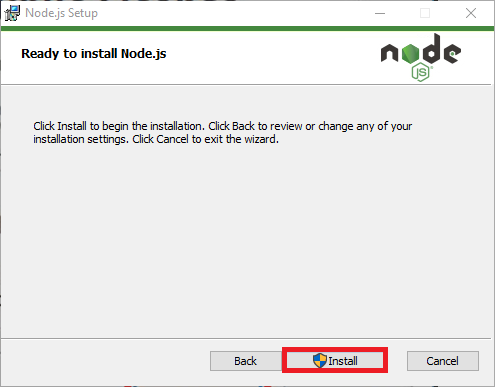


Figure 20: Begin installation. (Screen Capture of Node.js Setup)

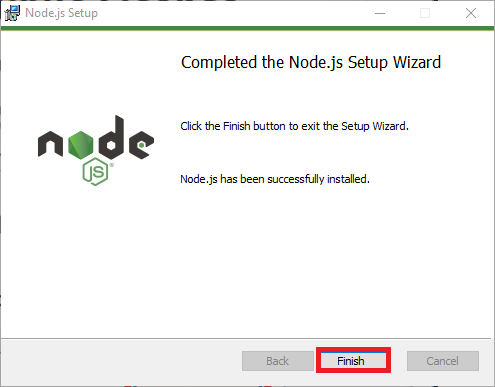


Figure 21: Finish installation. (Screen Capture of Node.js Setup)

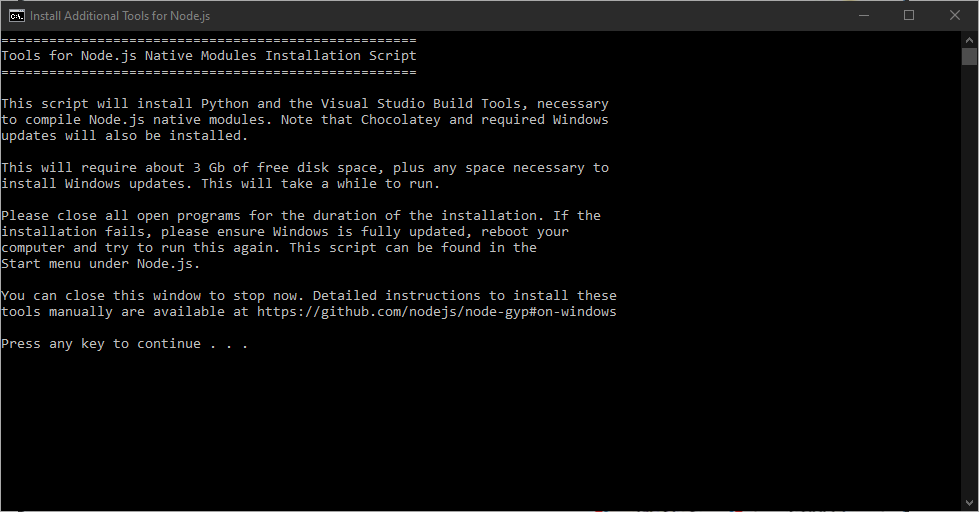


Figure 22: Install necessary tools by pressing any key (recommended). (Screen Capture of Install Additional Tools for Node.js)

Installing Angular CLI:

After Node.js is successfully installed, use the command from terminal to install the latest version of Angular CLI locally.

Installing npm packages:

After successfully installing the Angular CLI, use the command to install all the package dependencies locally.

Installing OpenLayer:

The final step is to install OpenLayers using the command*"* . You have now successfully setup the frontend.

## 3.4 Installing the backend

To install the backend database, navigate to the folder within the cloned repository folder. Download and install MongoDB 4.2 from <https://www.mongodb.com/try/download/community>, and follow the below steps to install:

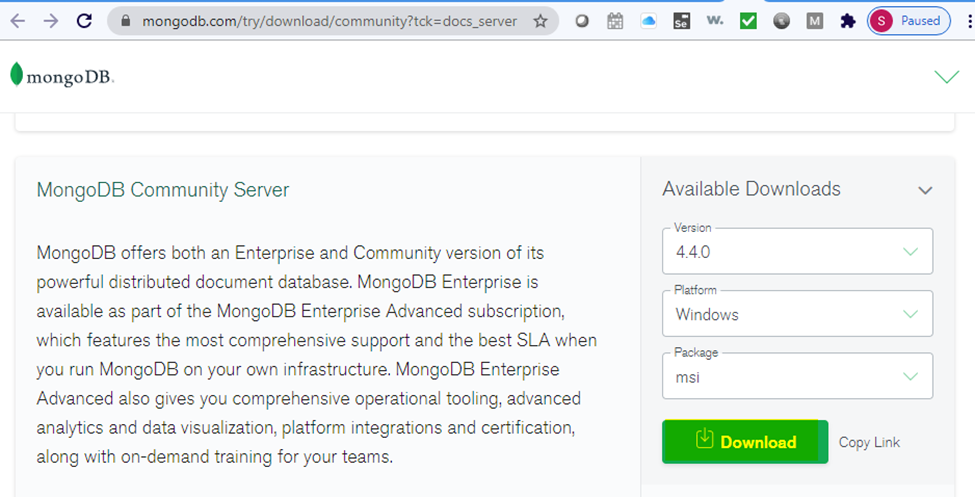


Figure 23: How to download MongoDB Community Edition

Installation steps:

1. In the Version dropdown, select the version of MongoDB to download.
2. In the Platform dropdown, select Windows.
3. In the Package dropdown, select msi.
4. Click Download.

Run the MongoDB installer. From the Windows Explorer/File Explorer:

1. Go to the directory where you downloaded the MongoDB installer (.msi file). By default, this is your Downloads directory.
2. Double-click the .msi file.

Follow the MongoDB Community Edition installation wizard. The wizard steps you through the installation of MongoDB and MongoDB Compass.

1. Choose Setup Type:

You can choose either the Complete (recommended for most users) or Custom setup type. The Complete setup option installs MongoDB and the MongoDB tools to the default location. The Custom setup option allows you to specify which executables are installed and where.

After successfully installing MongoDB 4.2, you will have to configure it using your local credentials. You have now successfully setup the backend.

## 3.5 Running the frontend

To run the frontend locally, use the command from within the folder. The application will automatically launch the system's default browser and navigate to <http://localhost:4200/>. The application will automatically reload if you change any of the source files. It is recommended to use the VS Code as the choice of IDE for modifying any of the source codes. It supports all the UI source file formats, including HTML, CSS, and TypeScript.

## 3.6 Running the backend

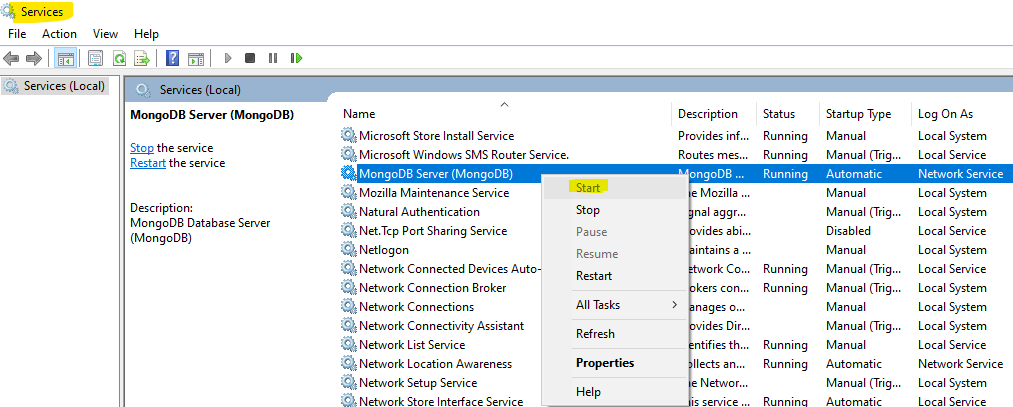
To run the backend locally, locate the MongoDB service in the service console, right-click the MongoDB service and click start

Figure 24: How to Start MongoDB Service

then connect a mongo.exe shell to the running MongoDB instance by opening a Command Interpreter with Administrative privileges and run:



Detailed steps on starting, stopping and uninstalling the backend database can be found in the "Municipality Permit Chatbot System Installation Guide" within the folder.

## 3.7 Running unit tests

To run the frontend unit tests, use the command from within the folder to start running the tests using Jasmine and Karma. The application will launch the tests in the default system browser and start running the existing tests. The tests will be rerun as you update or add any new tests.

## 3.8 Running end-to-end tests

To run the e2e tests, use the command from within the folder to start running the tests using Protractor.

# 4. Code structure

## 4.1 Frontend

### 4.1.1 Source code

All source code is located inside the folder. Follow the steps below to get the source code:

1. Navigate to [umgc.city\_chatbot](https://github.com/umgc/umgc.city_chatbot) GitHub repository

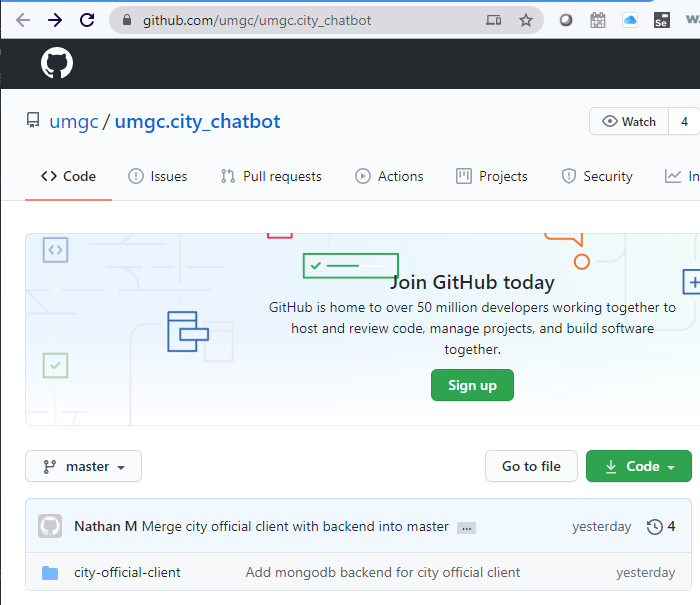


Figure 25: umgc.city\_chatbot GitHub repository

1. Navigate to city-official-client > scr > app

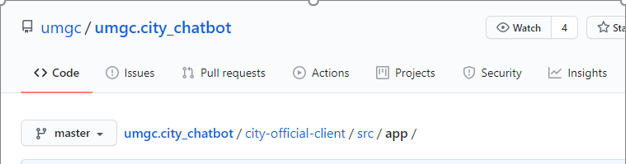


Figure 26: city-official-client folder

The list of all the current components are:

1. account
2. home
3. users
4. zone-dashboard

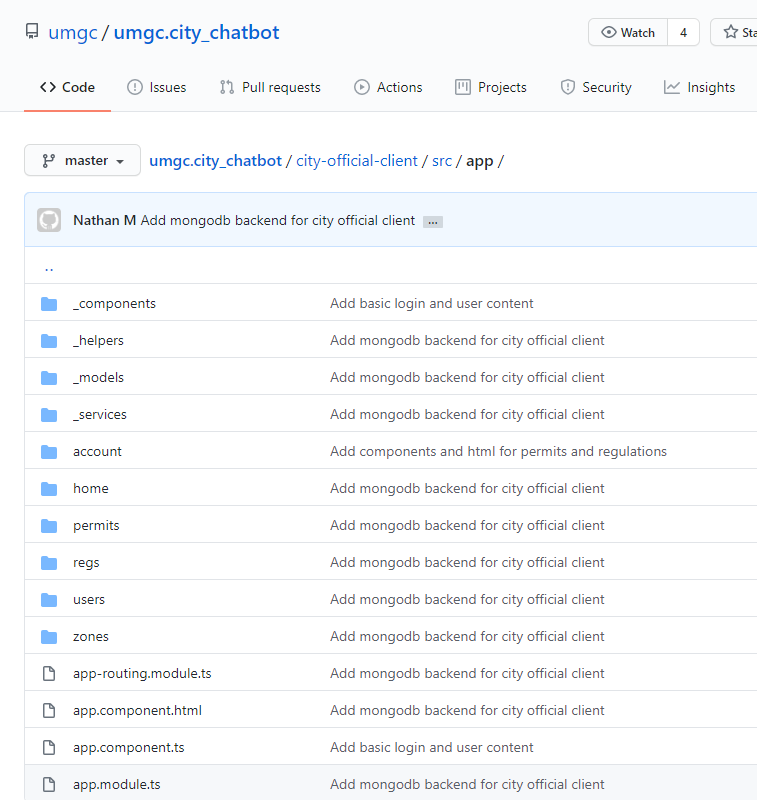


Figure 27: List of components in city-official-client folder

Each of these components contains a routing-module file for navigation from within the application, module file for including all the other components that this component can interact with, a component.html file for all the markup code, a component.ts file for all the code logic, and a spec.ts file for all the unit tests.

### 4.1.1.1 Account

The account component deals with user accounts and validation. It is the component that differentiates between admin users and other users.

### 4.1.1.2 Home

The home component is the landing page for the application. It contains the page that allows the users to enter their account id and password to authenticate and log into the system.

### 4.1.1.3 Users

The user component is used to retrieve the list of users in the system. It can also be used to add a new user to the system or remove an existing user from the system.

### 4.1.1.4 Zone-dashboard

The zone-dashboard component is the dashboard page the admin users see once they have been authenticated into the system. It allows admin users to be able to add, edit or delete zones, permits, and regulations stored in the system's database.

### 4.1.2 Assets

The assets folder contains all the files that are not limited to just one component but used throughout the application. Files that are usually included are images, JSON files, fonts, and top-level CSS.

### 4.1.3 Environments

The environments folder contains all the build configuration files. The environment variable and the system URLs are usually defined here. There are two different builds for the application, the development build used locally, and other integration environments and the production build used in UAT and production environments.

# 5. Code scaffolding

## 5.1 Frontend

Run to generate a new component. You can also use to generate the corresponding file.

The project utilizes the MEAN (MongoDB, Express.js, Angular, Node.js) stack. Changes to code are done on the angular frontend. Node, express, and MongoDB are utilized to supply a backend. Each datatype has its own angular component and service. Components make up the functionality and templating while services send data to the backend.

Changes can be seen in real time after starting Angular, a Nodemon server, and an instance of MongoDB.