Group 44 Product Documentation

CEN3031 - Introduction to Software Engineering

Joseph Somerdin- Project Manager
Ashish Satyavarapu- Scrum Master
Giovanni Cornejo- Development Team Member
Eric Clayton- Development Team Member

Product Vision Statement

Trash-Track is a waste control web application where users can view and add nearby trash and recycling bin locations. With every bin a user adds, they receive experience to level up their profile. Our motto is simple: Find nearby waste bins with one click. Wherever. Whenever.

Project Overview

Trash-Track is an app that was designed to eliminate the time spent searching for waste bins. No more worrying about where one can put their trash; our app is a quick way to locate the nearest trash can in seconds. We also support finding recycle bins. This app is an attempt to reduce pollution by giving people one less excuse to litter. Since finding nearby trash cans is as easy as clicking a button, no one should feel the need to litter. We also have a positive reinforcement system where users can earn experience points by adding bins to the database. These experience points correspond to levels that users can show off by sharing their profile link.

Requirements documentation

Trash-Track is an environmentally aware social media application that allows users to find or log nearby waste bins with the help of geolocation. The two core functionalities of our application are finding bins near a user and letting users log bins that have not already been stored in the database.

When opening the main page of the app without an account, users are greeted by a welcome page that describes the application's purpose along with a sign-up button to create an account with Trash-Track. Once a user hits the sign-up button, they are asked to use their Google account to sign in; this allows Trash-Track to remain secure by utilizing every layer of security that Google uses to protect a user's information. After linking a Google account, users are prompted to set up their account by inputting a unique username. Once the account is set up, the user is brought to their profile page where they can see stats such as experience points, user level, logged trash bins, and logged recycle bins. Users gain experience points when they add either new trash or recycle bins to the existing catalog of bins recorded in the database. Trash bins grant 10 experience points while recycle bins grant 20 experience points. Users can log out of their account by using the logout button. When the logout button is clicked, the page displays a message stating that the user is no longer logged in. Users can also edit their account using the settings button on their profile which allows them to modify their biography and profile picture. Users can also share profiles via a unique profile URL that allows anyone to view a user's stats, biography, and profile picture. Lastly, users can delete their account by clicking the "Delete Account" button on the settings page.

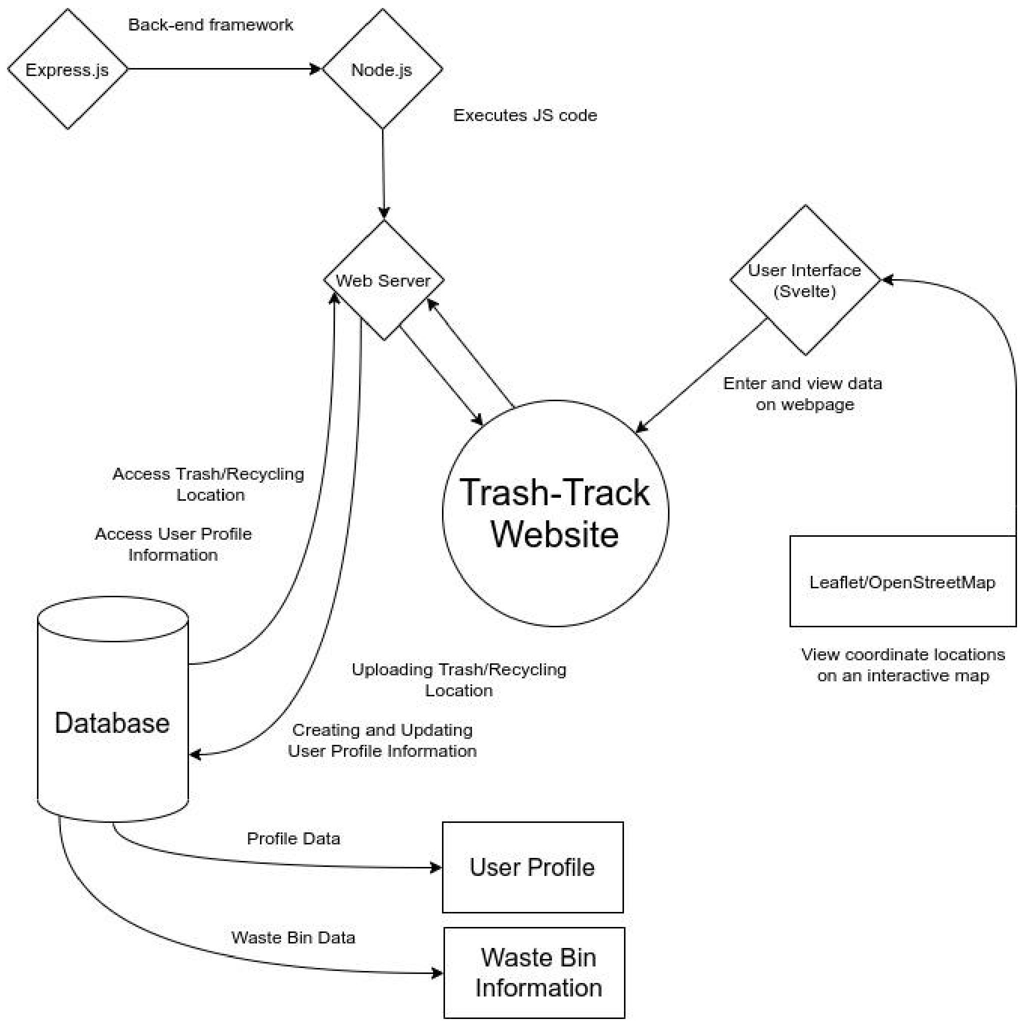
The bin finding and logging pages have several useful features that help make our app quick and easy to use. When inputting a location into the app, users can choose between three different methods depending on what is easiest for them. These methods include manually entering coordinate points, clicking on a map to fill in the coordinate points, or using HTML geolocation to find the user's current location. This should allow the app to be usable on any platform, not just mobile and not just desktop. Another useful feature that has been implemented into the app is the ability to click on a map marker and automatically open google maps with the selected location. The last useful feature is the ability to switch between trash and recycling bins by using a simple toggle switch.

Users can find nearby bins by navigating to the "Find" page. In order to search for nearby bins, the user will first input their coordinate position using the aforementioned methods. Once the coordinates and bin type are submitted, the application will take that input and make an API request that searches for the closest bin. Upon receiving a response, the app displays the bin on the map, the coordinates of the bin, and the distance from the user to the bin in meters. The application also notifies the user of who uploaded the bin that they searched for.

Users can add nearby bins by navigating to the "Add" page. After entering their coordinate position and bin type using the same methods as before, the app will make an API request to verify that the bin location is valid. The means for verification are as follows: no other bins can be within a 10 meter radius of any other logged bin of the same type and the user must not have submitted a bin within the last 10 minutes. The radius verification is used to ensure that there are no duplicate bins recorded and the rate-limit verification is used to ensure that users aren't spamming waste bins in order to falsely level up their profile. Once the API has verified that the location is valid, the database is updated to include the bin, recording its type, location, and the uploader's username.

System Documentation

The system architecture that our team has implemented is a Client-Server architecture. The server is responsible for hosting the API and serving the static content for the client frontend. The client makes http requests to the API which it then uses to dynamically update the view shown to the user on the website.



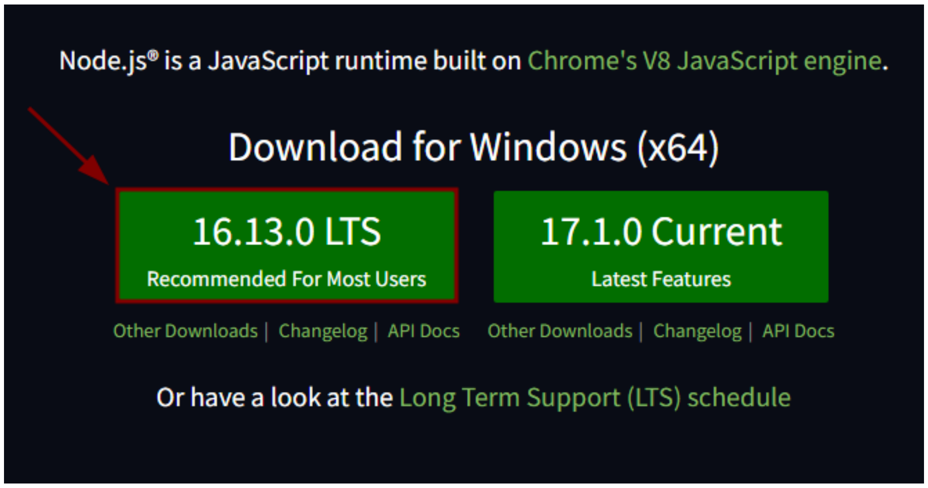
Back-end framework
Express.js
Node.js
Executes JS code
Web Server
User Interface (Svelte)
Enter and view data on webpage
Access Trash/Recycling Location
Access User Profile Information
Trash-Track Website
Leaflet/OpenStreetMap
View coordinate locations on an interactive map
Database
Uploading Trash/Recycling Location
Creating and Updating User Profile Information
Profile Data
User Profile
Waste Bin Data
Waste Bin Information

All components in our system context model revolve around the web server. Our team has used Express.js as a reliable back-end framework. This framework utilizes Node.js to run JavaScript code natively. Our front-end user interface is created with Svelte for users to enter and view data on the webpage. The Leaflet JS library and OpenStreetMap are used to display trash and recycling locations to the user through the use of markers. Our MongoDB database holds information including waste bin data and user profile data. The waste bin data includes location, bin type, and uploader username. The user data includes googleID, username, biography, profile picture URL, time last bin was added, logged trash bin count, logged recycle bin count, and total experience points.

Developer Documentation (Install/Build/Run Guide)

Install Required Dependencies

Windows:
Install the latest LTS version of Node.js from (https://nodejs.org/en/).
Keep all installation settings as default. Just click next until the installation is complete.

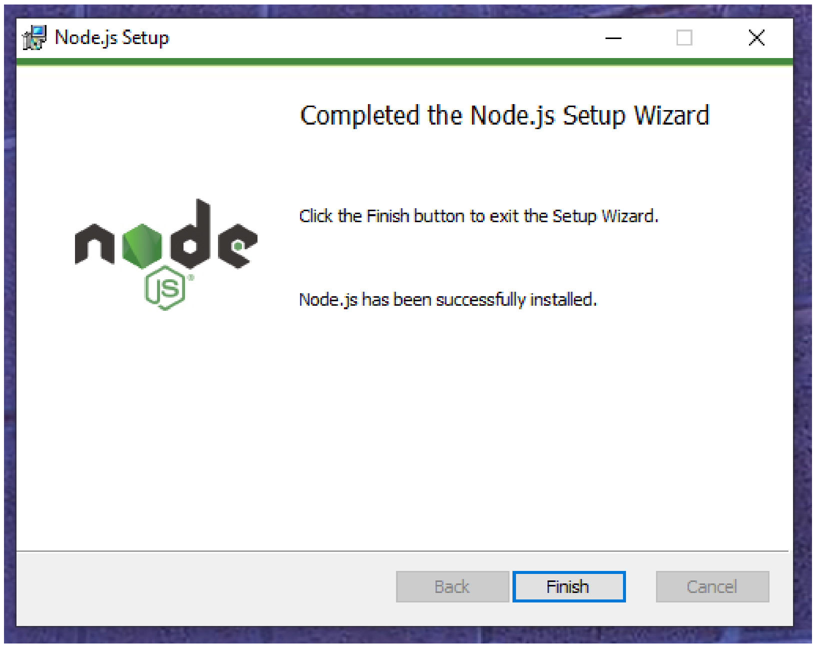


Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

Download for Windows (x64)

16.13.0 LTS
Recommended For Most Users
17.1.0 Current
Latest Features
Other Downloads | Changelog | API Docs
Other Downloads | Changelog | API Docs

Or have a look at the Long Term Support (LTS) schedule

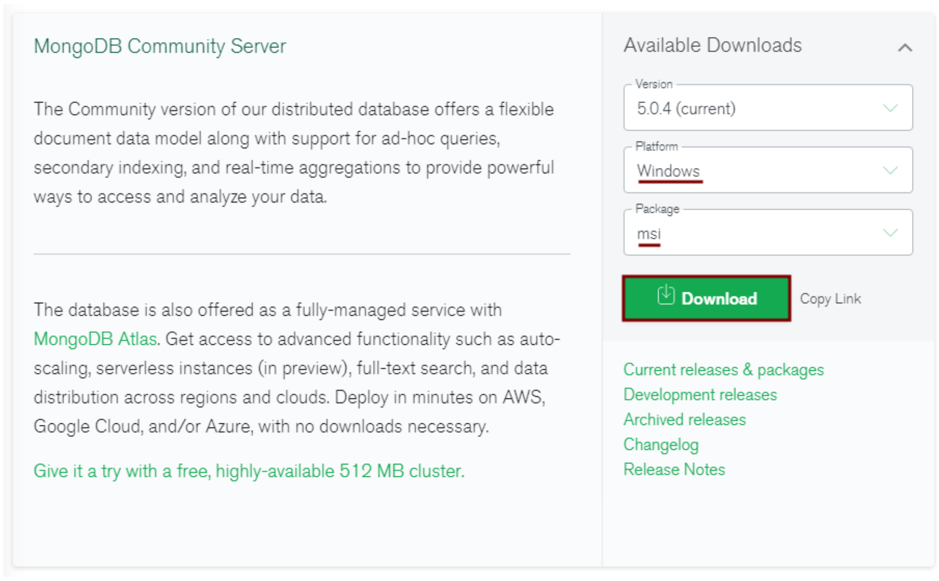


Completed the Node.js Setup Wizard

Click the Finish button to exit the Setup Wizard.

Node.js has been successfully installed.

Install MongoDB from (https://www.mongodb.com/try/download/community).
MongoDB Compass does not need to be installed.



MongoDB Community Server

The Community version of our distributed database offers a flexible document data model along with support for ad-hoc queries, secondary indexing, and real-time aggregations to provide powerful ways to access and analyze your data.

The database is also offered as a fully-managed service with MongoDB Atlas. Get access to advanced functionality such as auto-scaling, serverless instances (in preview), full-text search, and data distribution across regions and clouds. Deploy in minutes on AWS, Google Cloud, and/or Azure, with no downloads necessary.

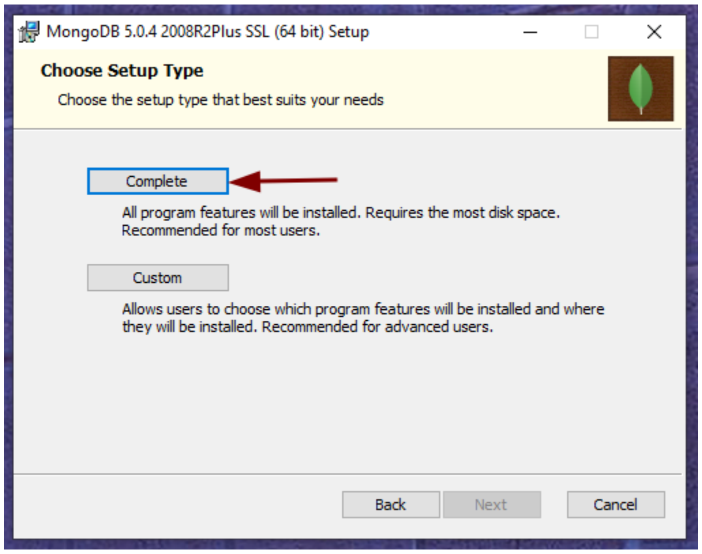
Give it a try with a free, highly-available 512 MB cluster.

Available Downloads

Version: 5.0.4 (current)

Platform: Windows

Package: msi



MongoDB 5.0.4 2008R2Plus SSL (64 bit) Setup

Choose Setup Type

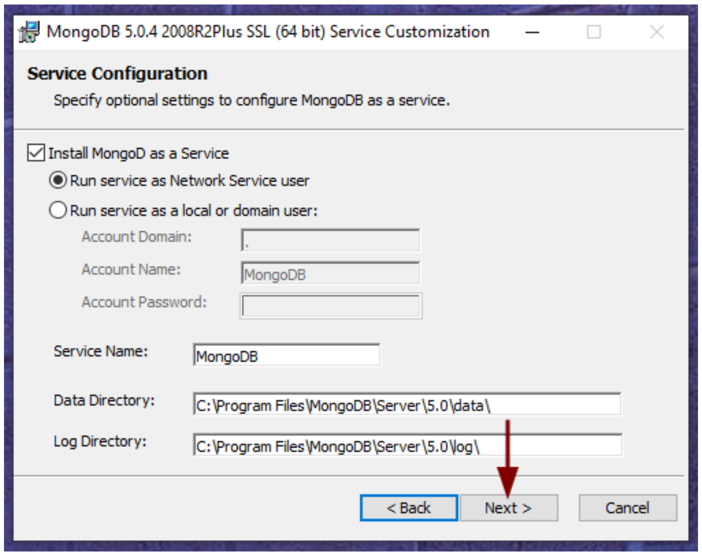
Choose the setup type that best suits your needs

Complete

All program features will be installed. Requires the most disk space. Recommended for most users.

Custom

Allows users to choose which program features will be installed and where they will be installed. Recommended for advanced users.



MongoDB 5.0.4 2008R2Plus SSL (64 bit) Service Customization

Service Configuration

Specify optional settings to configure MongoDB as a service.

☑ Install MongoD as a Service

● Run service as Network Service user

○ Run service as a local or domain user:

Account Domain: .

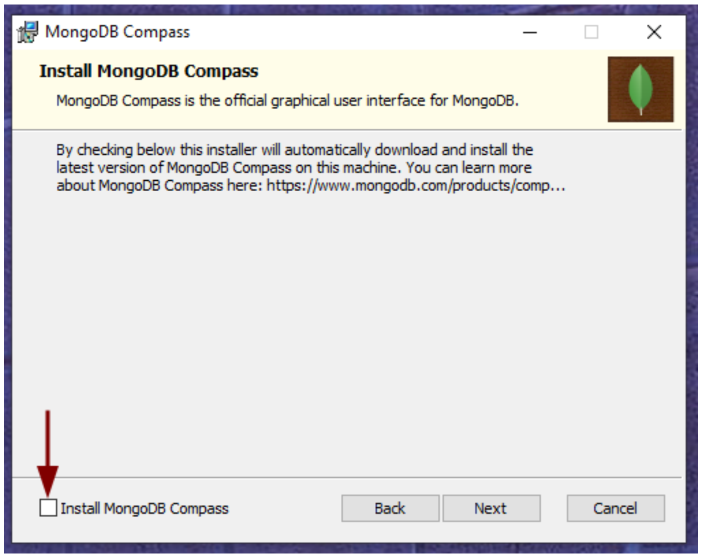
Account Name: MongoDB

Account Password:

Service Name: MongoDB

Data Directory: C:\Program Files\MongoDB\Server\5.0\data\

Log Directory: C:\Program Files\MongoDB\Server\5.0\log\



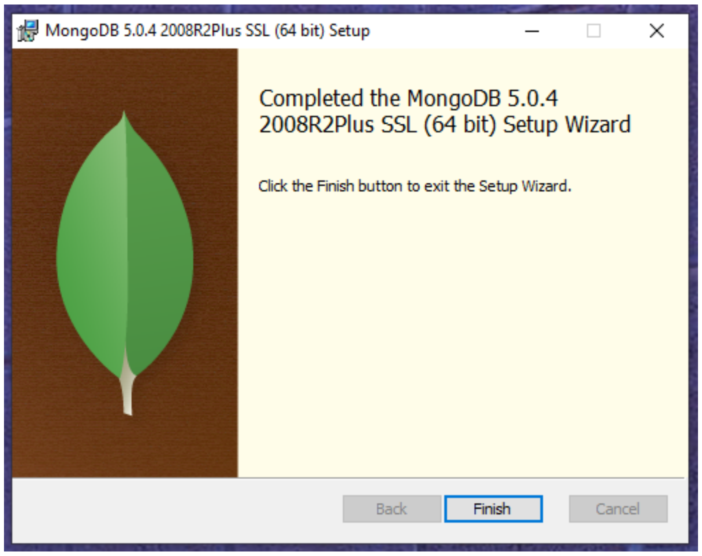
MongoDB Compass

Install MongoDB Compass

MongoDB Compass is the official graphical user interface for MongoDB.

By checking below this installer will automatically download and install the latest version of MongoDB Compass on this machine. You can learn more about MongoDB Compass here: https://www.mongodb.com/products/comp...

☐ Install MongoDB Compass



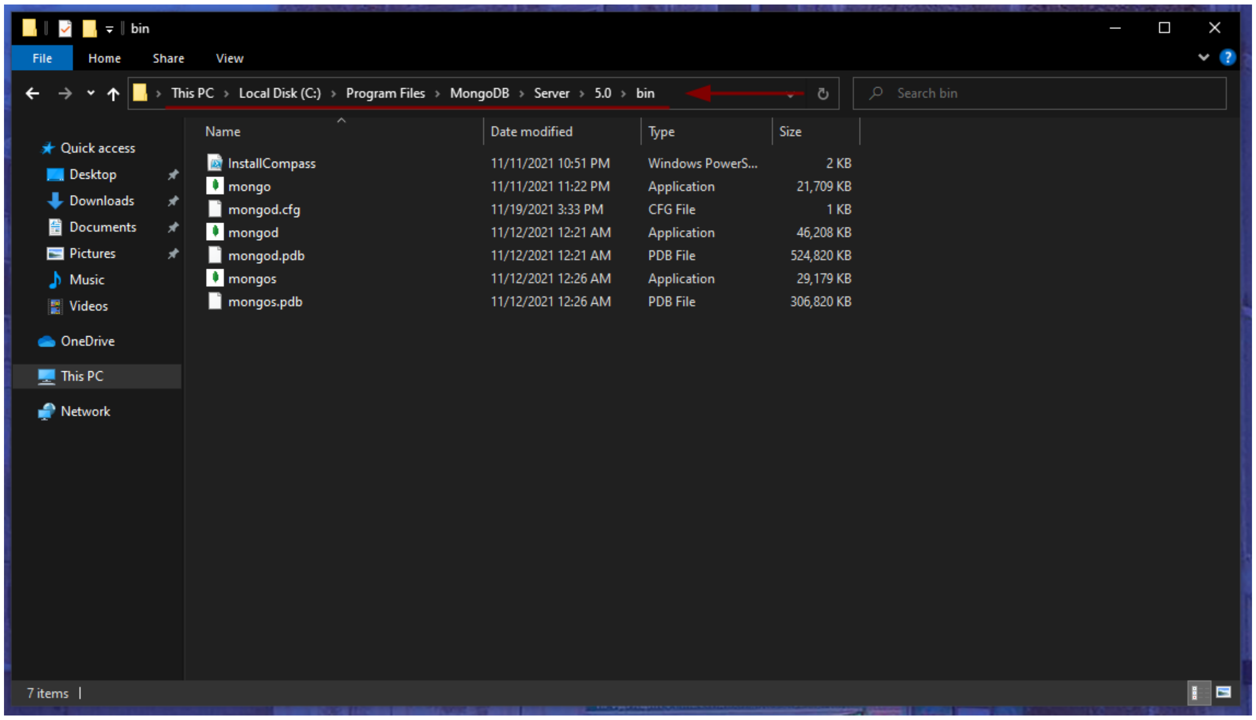
MongoDB 5.0.4 2008R2Plus SSL (64 bit) Setup

Completed the MongoDB 5.0.4 2008R2Plus SSL (64 bit) Setup Wizard

Click the Finish button to exit the Setup Wizard.

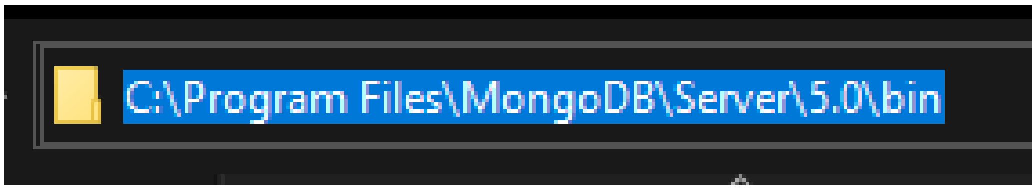
Add the installation directory of MongoDB to the user path by doing the following.

Navigate to the MongoDB installation directory.

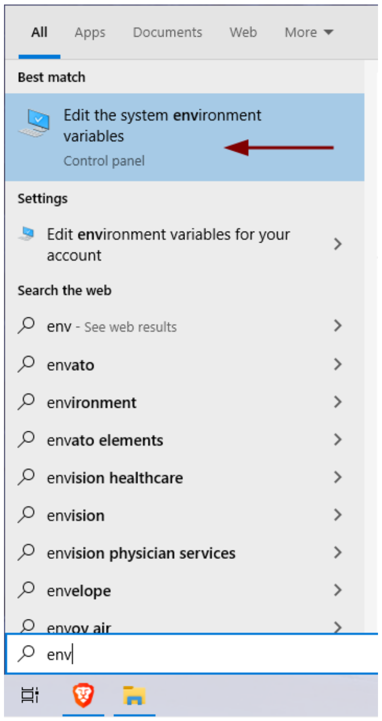


Name	Date modified	Type	Size
InstallCompass	11/11/2021 10:51 PM	Windows PowerS...	2 KB
mongo	11/11/2021 11:22 PM	Application	21,709 KB
mongod.cfg	11/19/2021 3:33 PM	CFG File	1 KB
mongod	11/12/2021 12:21 AM	Application	46,208 KB
mongod.pdb	11/12/2021 12:21 AM	PDB File	524,820 KB
mongos	11/12/2021 12:26 AM	Application	29,179 KB
mongos.pdb	11/12/2021 12:26 AM	PDB File	306,820 KB

Copy the MongoDB installation path to your clipboard.

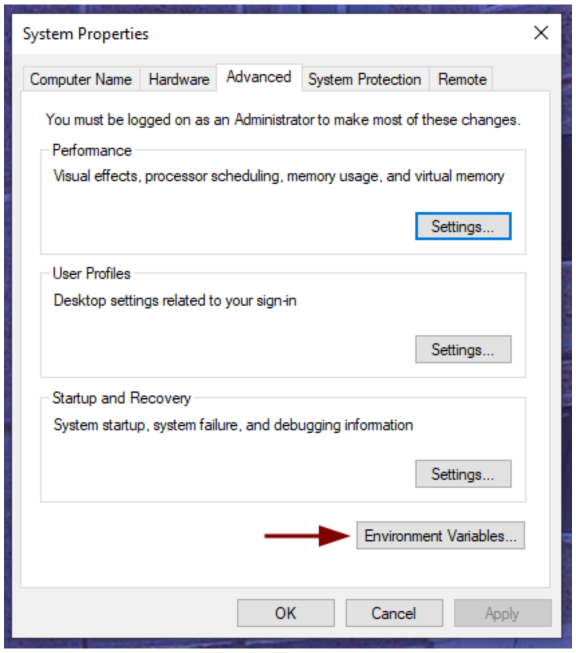


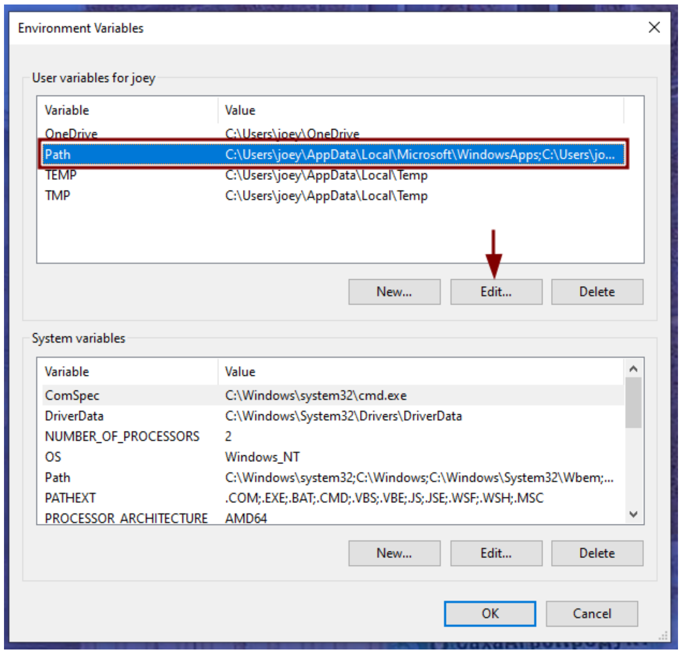
C:\Program Files\MongoDB\Server\5.0\bin



Type “env” into the start menu and click the first result.

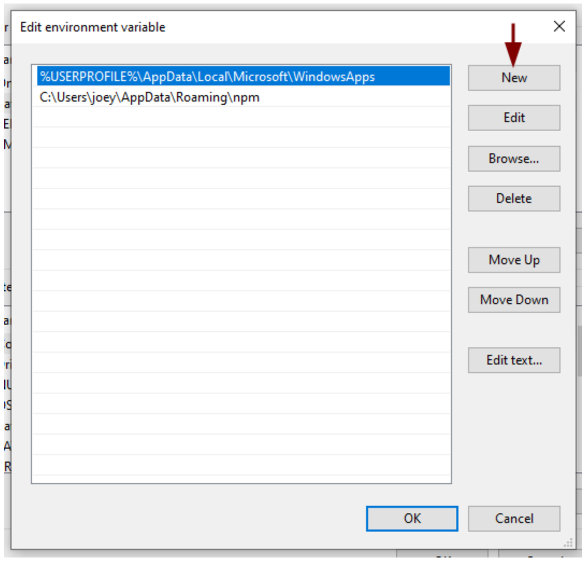
Follow the red arrows, paste in the MongoDB installation path, then click ok.


System Properties
Computer Name
Hardware
Advanced
System Protection
Remote
You must be logged on as an Administrator to make most of these changes.
Performance
Visual effects, processor scheduling, memory usage, and virtual memory
Settings...
User Profiles
Desktop settings related to your sign-in
Settings...
Startup and Recovery
System startup, system failure, and debugging information
Settings...
Environment Variables...
OK
Cancel
Apply

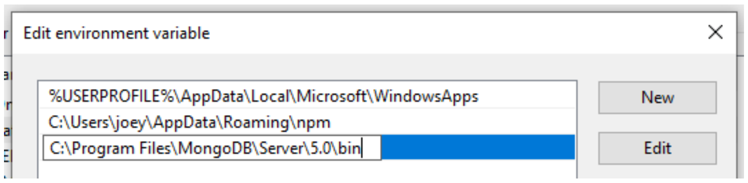


Variable	Value
OneDrive	C:\Users\joey\OneDrive
Path	C:\Users\joey\AppData\Local\Microsoft\WindowsApps;C:\Users\jo...
TEMP	C:\Users\joey\AppData\Local\Temp
TMP	C:\Users\joey\AppData\Local\Temp

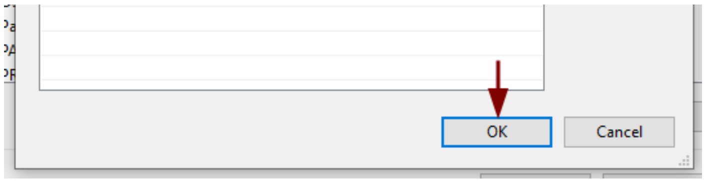
Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	2
OS	Windows_NT
Path	C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64



%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Users\joey\AppData\Roaming\npm



%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Users\joey\AppData\Roaming\npm
C:\Program Files\MongoDB\Server\5.0\bin



Mac:

Install Homebrew by following the instructions at (https://brew.sh/).
After brew is installed run the following commands in a new terminal window:

```
brew install node
brew tap mongodb/brew
brew install mongodb-community
```

Linux:

Arch based systems:

```
sudo pacman -Syy
sudo pacman -S nodejs npm
yay -S mongodb-bin
```

For any other distribution, use your package manager to install NodeJS, npm, and MongoDB.

Build and Run (All Platforms)

Clone the repository:

```
git clone https://github.com/asleeponpluto/11282group44.git
```

Install node dependencies:

```
cd 11282group44/client
npm install
cd ../server
npm install
```

Create a new file in the server directory with the name ".env":

```
# just copy and paste the following into the file and save

GOOGLE_CLIENT_ID=334333330406-c44nu8s6q536udjq97nn5al04bmef20s.apps.googleuse
rcontent.com
GOOGLE_CLIENT_SECRET=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
```

Build Svelte frontend:

```
# must be in client directory
npm run build
```

Run server:

```
# must be in server directory
npm run start
```

Now navigate to `http://localhost:8080` in your browser to use the application.