All files that were created / edited by me for this project are:  
  
Everything inside \backend folder (excluding package.json files)

Everything inside \frontend\src, the other files in frontend are VUE config files

Problems encountered

The only real problem that I encountered during development of this project was the rendering of the chat on the frontend. I spent so long attempting to create an algorithm that would work out the space needed between the user and agent chats based on the number of lines in each, but eventually I had to abandon this as it was taking too much time, and I only had a set amount of time for this project. Instead, I duplicated each message sent, but with the duplicated message having a “hidden” class. This would hide the message, leaving the correct amount of space between the chat logs.

Limitations

A limitation of my design is that it utilises a frontend and a backend, which means there is twice the amount of services to manage. This may mean that the hardware requirements for running the software could be slightly greater than just a standalone application. There are also some security concerns that come into play when using databases such as MongoDB, but as we’re only keeping the data temporarily, and none of it is sensitive data the security concerns are less so.

Future Improvements

One future improvement that could be made is that the frontend could be packaged into a VueJS component, which would easily allow it to be deployed onto another site.

With the frontend being made with the VueJS framework, then it is also easily deployed as standalone iOS and Android apps, as long as the backend is hosted somewhere.

More questions could be added to the agent which would allow the filtering of holidays by fields such as country and star rating.

The styling / UI could be improved to be a little less simple.

Conclusion

To conclude this project, I believe that what I have delivered meets all the requirements set at the beginning. I think this was a simple, yet challenging in some areas, project that I managed to complete using all my knowledge learnt from my apprenticeship. I think that the results of the chat agent are (in my opinion) exactly what was asked for. The most difficult part of the project was definitely trying to figure out the chat rendering, but once I’d got that sorted everything just fell into place.

I think this project neatly sums up everything that I’ve learnt over the last 18 months

User Guide

PREREQUISITES

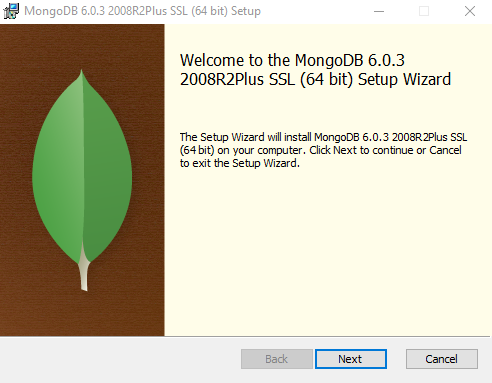
Downloading

1. download MongoDB from https://www.mongodb.com/try/download/community

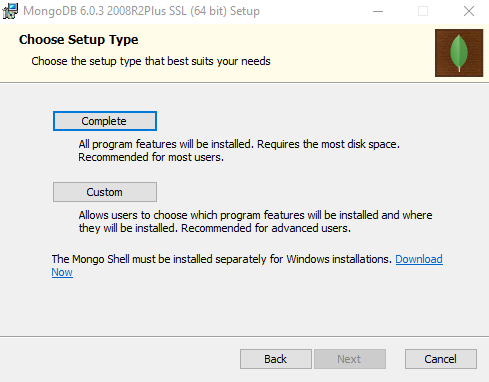
2. download NodeJS from https://nodejs.org/dist/v18.12.1/node-v18.12.1-x64.msi

Installing

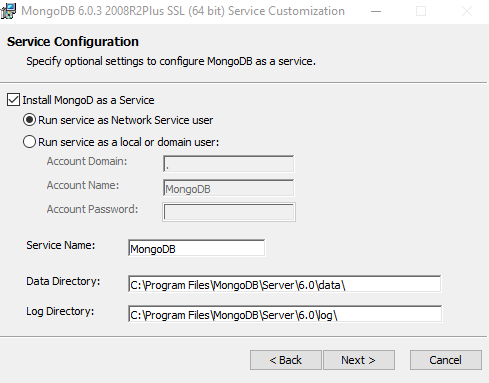
1. Install MongoDB using the installer:



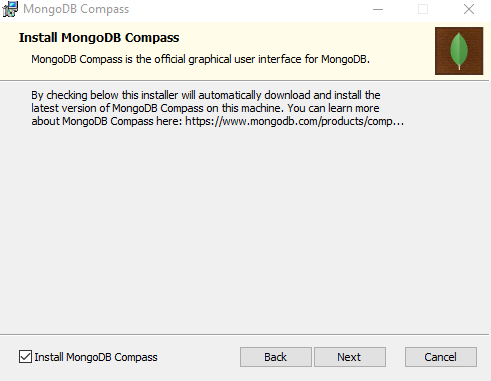
Choose Complete



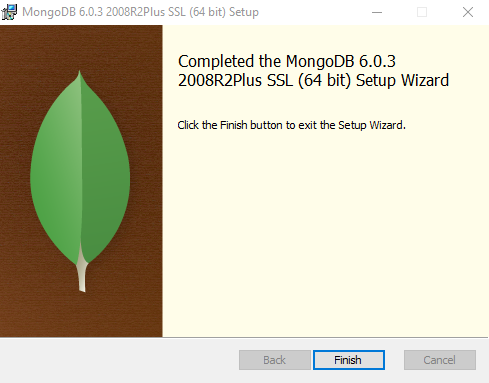
Don’t change anything here, just press next



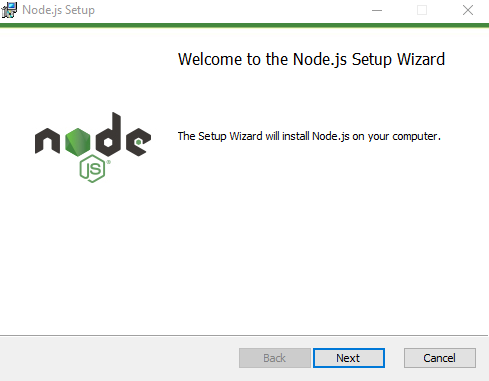
Ensure the “Install MongoDB Compass” is ticked



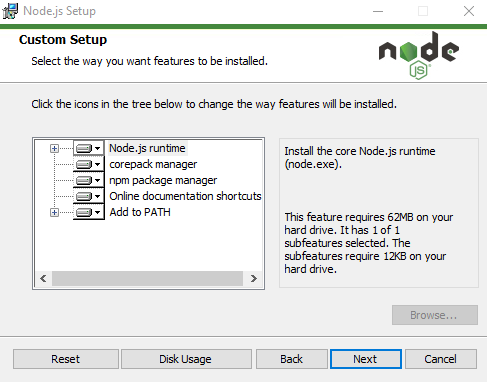
Wait for the install to finish



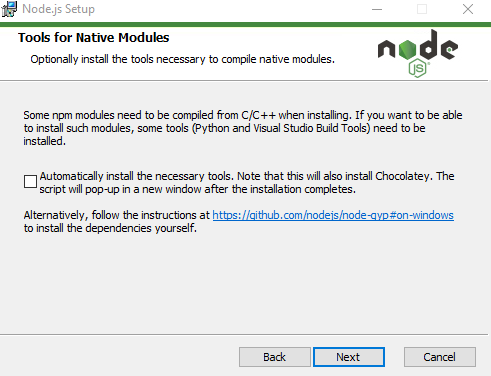
2. Install NodeJs using the installer:



Don’t change anything here, just press next:



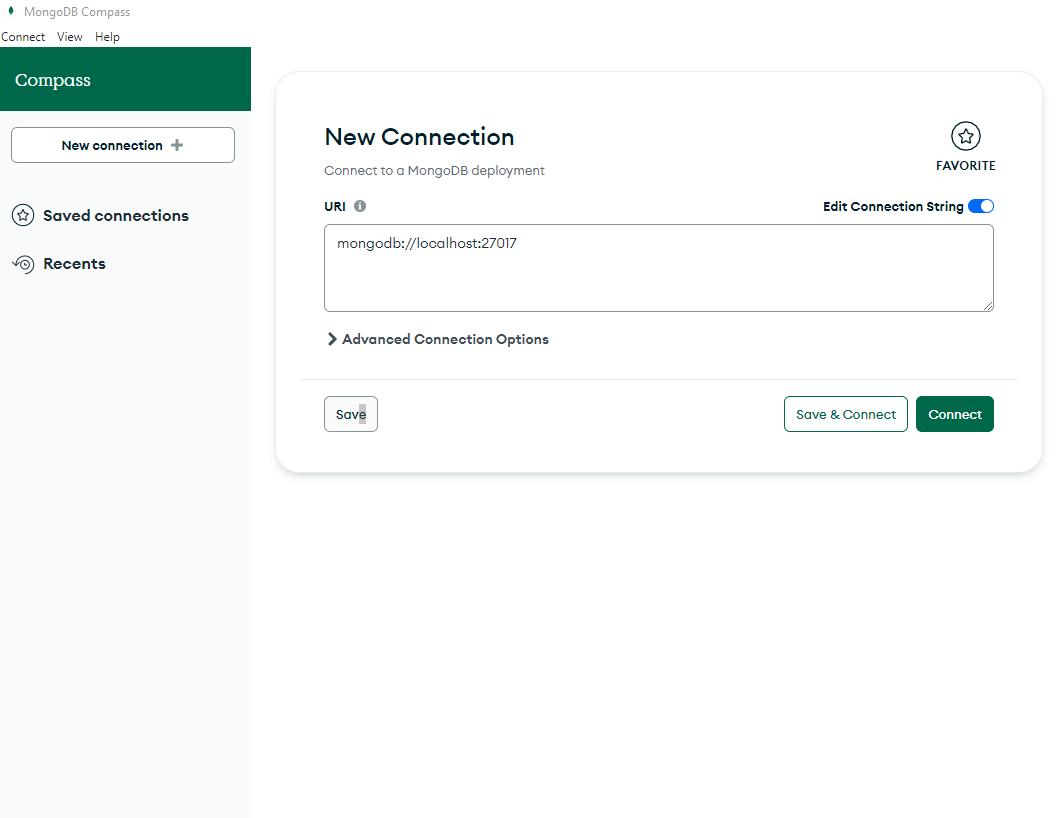
Ensure the “Automatically install the necessary tools” box is un-ticked



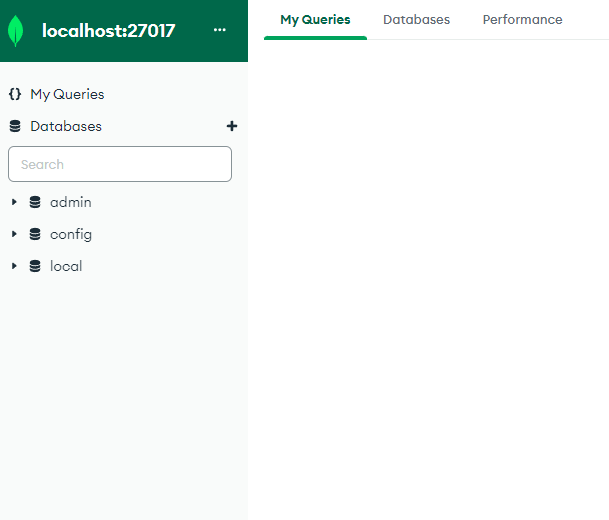
And then wait for the setup to finish

General

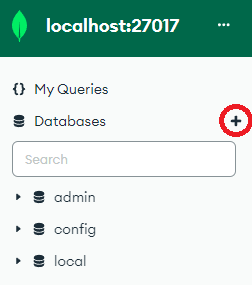
1. Check that MongoDB server is installed and running using MongoDBCompass

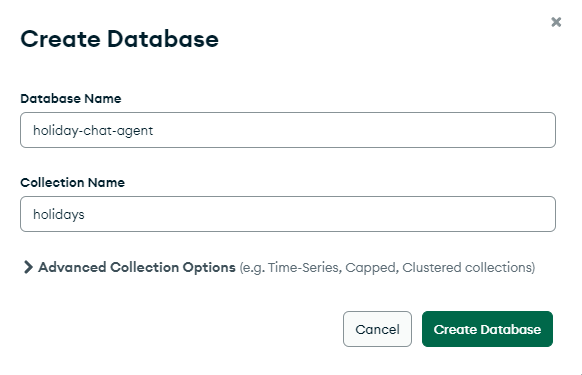


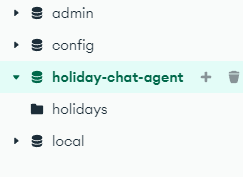
Hit “connect” and then you should see this in the top left corner



2. Create a new database with MongoDBCompass called "holiday-chat-agent" with a collection called "holidays"

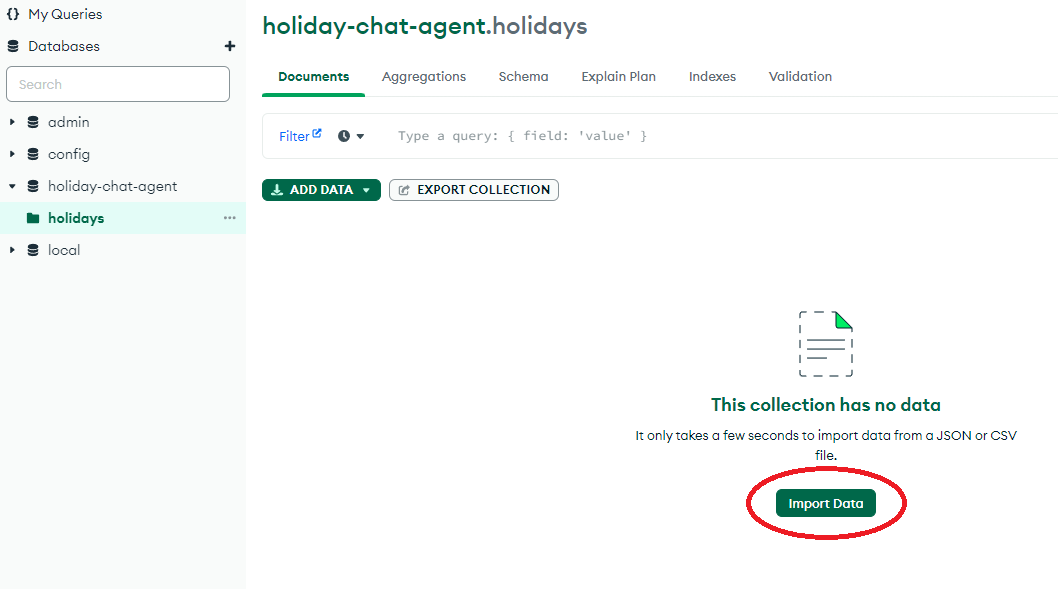


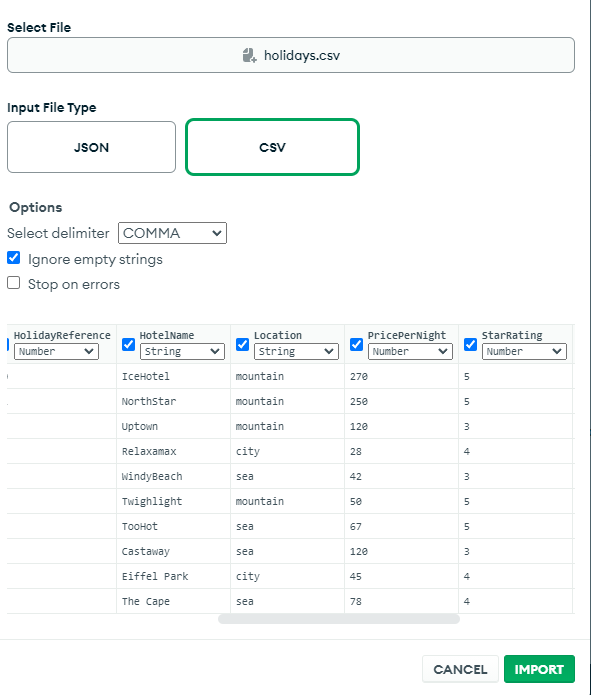




3.In the holidays collection, import the holidays.csv file containing all the holiday data

(Ensure file type is CSV, ignore the rest of the options)



Ensure that HolidayReference, HotelName and PricePerNight are all “Number” rather than “String

4. Check that NodeJS is installed by opening command prompt and entering "node -v"

START THE APP

1. Open a command prompt and navigate (cd) to the HolidayChatAgent/backend folder

2. Use the "npm i" command to install all node dependencies

3. Use the "nodemon" command to start the server

Upon server start cmd should print:

**[nodemon] starting `node index.js`**

**Server is listening on port: 3000**

**Connected to the Database successfully**

4. Open another command prompt and navigate (cd) to the HolidayChatAgent/frontend folder

5. Use the "npm i" command to install all node dependencies

6. Use the "npm run serve" command to run the frontend server

7. Open a browser and connect to the frontend with the url of http://localhost:8080/

USER GUIDE

Using the app itself it is quite intuitive, the agent will direct you in which keyword to say to start the questions, after all questions have been successfully answered the agent will thank you for using the service and direct you to a start again button.