

**Lucca Marle**

Started**RAG to Riches: Build your first AI chatbot**



Lucca Marle

11:52 AM

Hey there!

11:52

Before we proceed, what would you like your preferred nickname to be from now on?



You

11:53 AM

Vrishali Sanglikar



Lucca Marle

11:53 AM

Cool! I'll use **Vrishali Sanglikar** from now on.

11:53

[](https://media.giphy.com/media/udfgxip30ltcgDxuZX/giphy.gif)

11:53

Hi, I'm **Lucca** and I work at Anythink, your virtual company for this quest.

11:53

We use Anythink in this quest and on Wilco for simulating real-life scenarios and much more.

11:53

Let's get started.

11:53

Let’s select your preferred tech stack that is available

11:53

We currently have this single backend framework supported for this task. Please click to proceed.

Top of Form

* node

Bottom of Form



Lucca Marle

11:53 AM

Excellent choice!

11:54

Let me set your personal GitHub repo...

11:54

We need you to connect using your GitHub account.

11:54

Let's do it by clicking connect and approve our app.

11:54

Connect to Github



Lucca Marle

12:12 PM

Great we have everything we need so far, I've created a repo for you and added you to it, this is your [repository](https://github.com/Wilcolab/Anythink-Market-uj0hvait)

12:12

We are going to use MongoDB Atlas as our database.

12:12

MongoDB Atlas is a cloud-hosted database service that allows you to get started in minutes.

12:12

We will be using the free tier, which is more than enough for our needs.

12:12

To get started, go to the [MongoDB register page](https://www.mongodb.com/cloud/atlas/register) and create an account.

12:12

Choose the Cluster Tier of M0 – the free tier, and pick your preferred cloud provider and region. You can also choose to keep the default settings

12:12

Ready?

Top of Form

* I'm ready
* Not yet

Bottom of Form



Lucca Marle

12:12 PM

Great, your cloud database will be ready in no time!

12:12

Make your way to either Security > Quickstart or Network Access on the left menu.

12:12

You'll assign a unique username and a secret code. Remember to save these credentials for later.

12:13

To ensure our app can access the database, allow all IPs to access it (this can be changed later if you wish to strengthen its security).

12:13

Add the IP 0.0.0.0/0 to the IP Access List

12:13

Ready?

Top of Form

* I'm ready
* Not yet

Bottom of Form



Lucca Marle

12:13 PM

Perfect! Your MongoDB Atlas cluster is now accessible for your app.

12:13

Now that you have your MongoDB Atlas cluster set up, we need to configure the connection string. A connection string is a unique address that tells your app where to find the database.

12:13

Head over to Deployment > Database.

12:13

Spot your new cluster, and click on the Connect button.

12:13

Following this, you'll find MongoDB for VS Code. It should resemble the following example: mongodb+srv://<username>:<password>@<cluster>.mongodb.net/?retryWrites=true&w=majority.

12:13

Copy the connection string and paste it here.

Make sure that you update the connection string with the username and password you created earlier.

12:13

The cluster creation process can take a few minutes. If you get an error, wait a few minutes and try again.

12:13

Submit



Lucca Marle

12:23 PM

That's not correct, we received an error . The initial load of the cluster might take a few minutes. Try again in a bit.



Lucca Marle

12:25 PM

That connection string looks good! We are ready to move on to the next step.

12:25

To empower our AI assistant, we've loaded movies data for you. Let's delve into understanding embeddings and vectors.

12:25

Various tools like Atlas Web UI, MongoDB Compass, MongoDB Shell, or alternate MongoDB clients can be used to explore the data. For now, we'll focus on the Atlas Web UI.

12:25

Navigate to Deployment > Database > Browse Collections in the Atlas Web UI.

12:25

Within the movies\_quest database, explore the embedded\_content collection.

12:25

Each document now contains a field named embedding a result of applying the text-embedding-ada-002 model to the plot field.

12:25

Think of the embedding model as a function; it takes text as input and returns a vector—a numerical representation in a multi-dimensional space. This representation captures semantic relationships, enabling tasks like sentiment analysis.

12:25

Vectors have fixed dimensions, reflecting the number of features. For instance, a 3-dimensional vector has 3 elements.

12:25

In the database, vectors are represented as arrays of numbers, where the array size corresponds to the dimensionality of the vector.

12:25

Can you determine the number of elements (known as dimensionality of the vector) in the embedding field in the database?



You

1:08 PM

1536



Lucca Marle

1:08 PM

Excellent! Each vector has 1536 dimensions, and we'll leverage these vectors to identify the requested movies.

1:08

For our AI assistant to work, we need to create an index on the embedding field. Let's use the Data Explorer in the Atlas Web UI to create the index.

1:08

Navigate to your cluster if you're not already there, click on Atlas Search tab, then Create Search Index, then under Atlas Vector Search choose the JSON Editor. Make sure you choose Vector Search index and not any other type

1:08

In the Database and Collection section, choose the movies\_quest database and the embedded\_content database.

1:08

In the dialog, enter the embedding field, its dimensions, and the similarity function

1:08

You can keep the default index name as vector\_index. Here is the complete JSON config for the index

1:09

{ "fields": [ { "numDimensions": 1536, "path": "embedding", "similarity": "cosine", "type": "vector" } ] }

1:09

Ready?

Top of Form

* I'm ready
* Not yet

Bottom of Form



Lucca Marle

1:13 PM

Perfect. We shall now be able to search for movies using the embedding field efficiently.

1:13

Before we continue, please [**open your new Codespace dev environment**](https://app.wilco.gg/my-codespace) or click on the Snack side menu.

1:13

Open the Codespace terminal and run docker compose up to get your app running. **Tell me when everything is running so we can continue**

1:13

Ready?

Top of Form

* I'm ready
* Not yet

Bottom of Form



Lucca Marle

1:19 PM

Cool.

You can go to the “Ports” tab in your Codespaces instance and see the url of your running app

1:19

Our chatbot is almost ready! We need to set the environment variables for the chatbot to work.

1:19

Head over to the backend folder and create an .env file

This can be done using the command

cd backend && touch .env

1:19

In the .env file, add the following variables,VECTOR\_SEARCH\_INDEX\_NAME=vector\_index MONGODB\_DATABASE\_NAME=movies\_quest MONGODB\_CONNECTION\_URI=your-connection-string

1:19

Update the MONGODB\_CONNECTION\_URI variable with your connection string

1:19

Once you're done, run docker compose up again to start the backend and the frontend with the updated environment.

1:19

You can now start chatting with the chatbot! Let's test it out.

1:19

Ask it what is the name of the movie with a storyline set on the planet "Mongo"

1:19

Which movie did it return?



You

1:27 PM

Flash Gordon



Lucca Marle

1:27 PM

That's spot on, our chatbot is ready to go!

Finished**RAG to Riches: Build your first AI chatbot**



Lucca Marle

1:27 PM

Great job! 😀 We still have a lot of work ahead of us, so [click here](https://app.wilco.gg/home) once you're ready for your next challenge, ok?

Top of Form

Bottom of Form

**Lucca Marle** is typing

···

**Current Quest:** Never Build Permissions Again