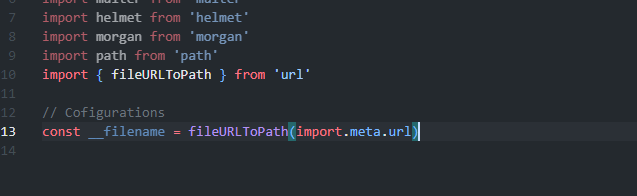
# **Configurations Done**

## Import.meta

When using modules in node js you need to create a object called import.meta in which you will create a property called URL in which you will have the url to the file

Note: this is done only when you are using modules



## FileURLToPath

This URL.fileURLToPath function decodes the file URL to a path string and ensures that the URL control characters (/, %) are correctly appended/adjusted when converting the given file URL into a path.

## Helmet (Security)

Helmet is used for the security purpose. It secures your header with different properties eg hiding the framework it has used to create a website eg express so this info cannot be used to attack on the site.

DNS Prefetch control is set to off.

By default DNS fetches all the links in your website even before the user clicks on the link but with helmet turning it off it is not done.

## CORS (Security)

Cors is set to cross origin using the helmet property which allows the frontend to consume the API.



## Morgan (logging)

Morgan is used for logging purpose it logs all the route the site has taken to complete the request.

There are different types a log is printed out of one is common which is used in the project.

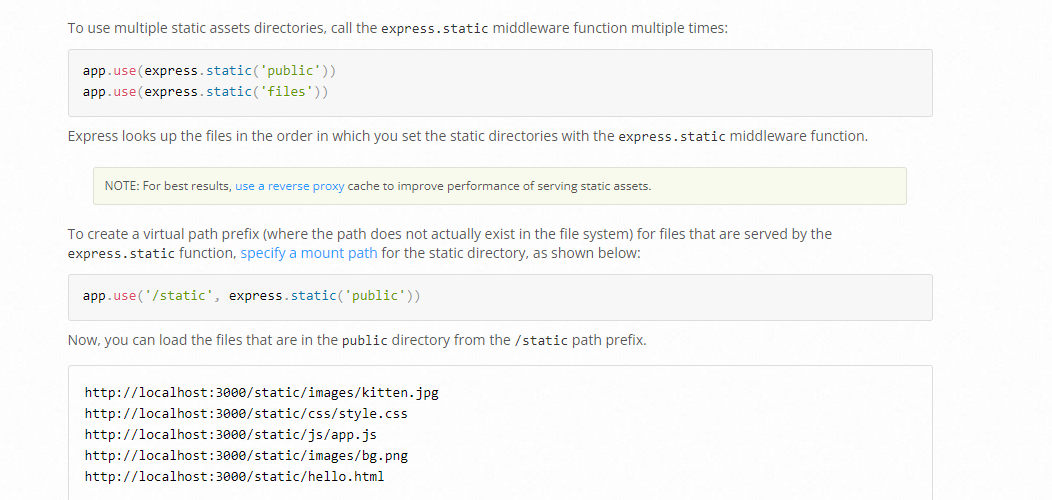
## BodyParser

Parses the incoming request and creates a json of body with the property name as body.

This can be accessed using req.body

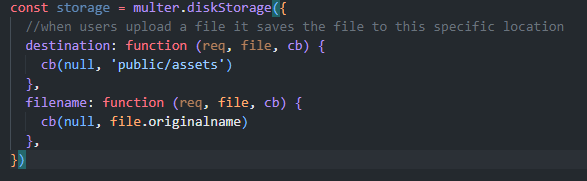
## Express.static

This is to load the static pages such as images, css file, javascript



## Multer (Storage)

Allows you to store a uploaded file to specific location

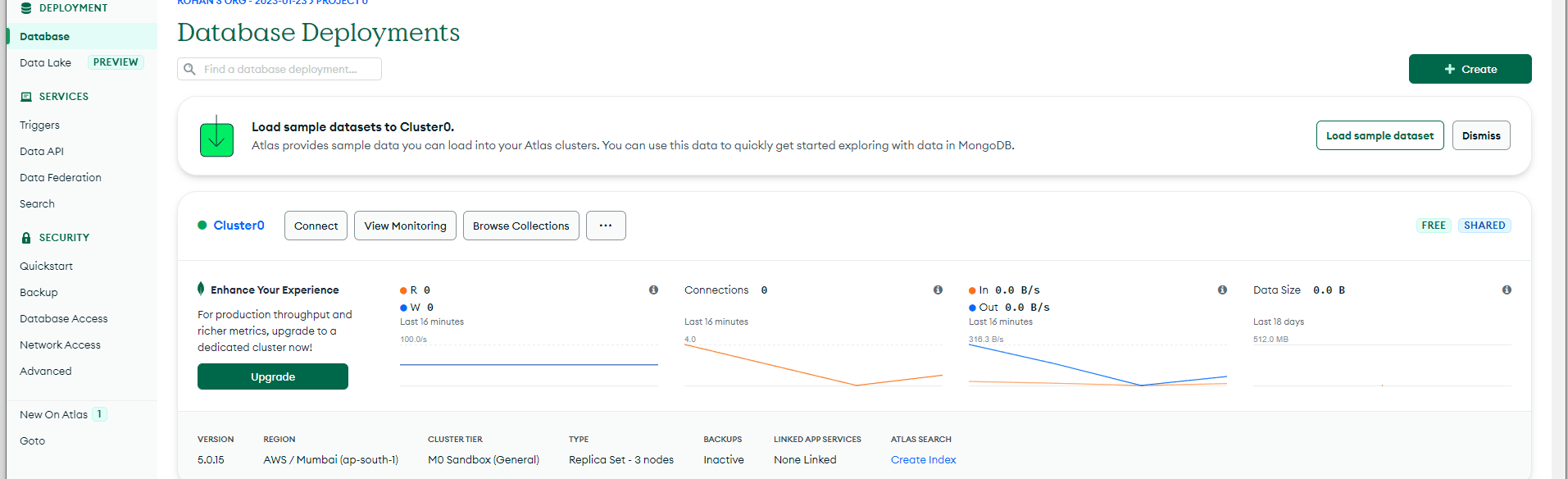


# Database

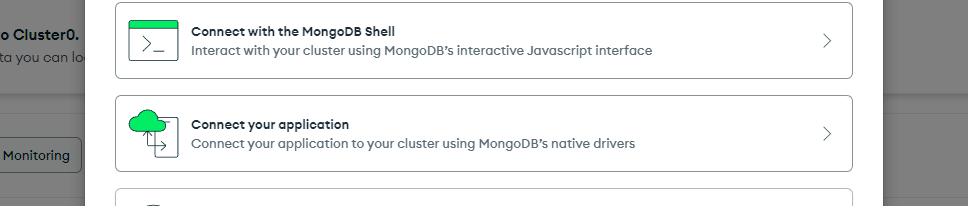
Database used is mongoDB

Steps to create and connect to mongoDB database

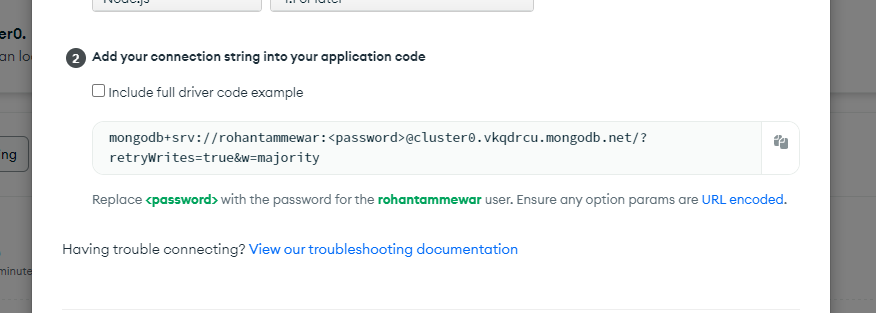
1. Create a database in mongodb site
2. In database tab click on connect



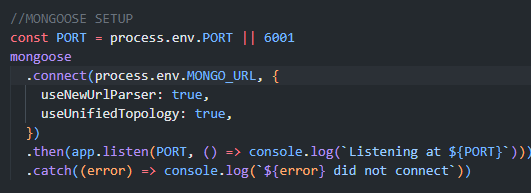
1. Click on connect your application



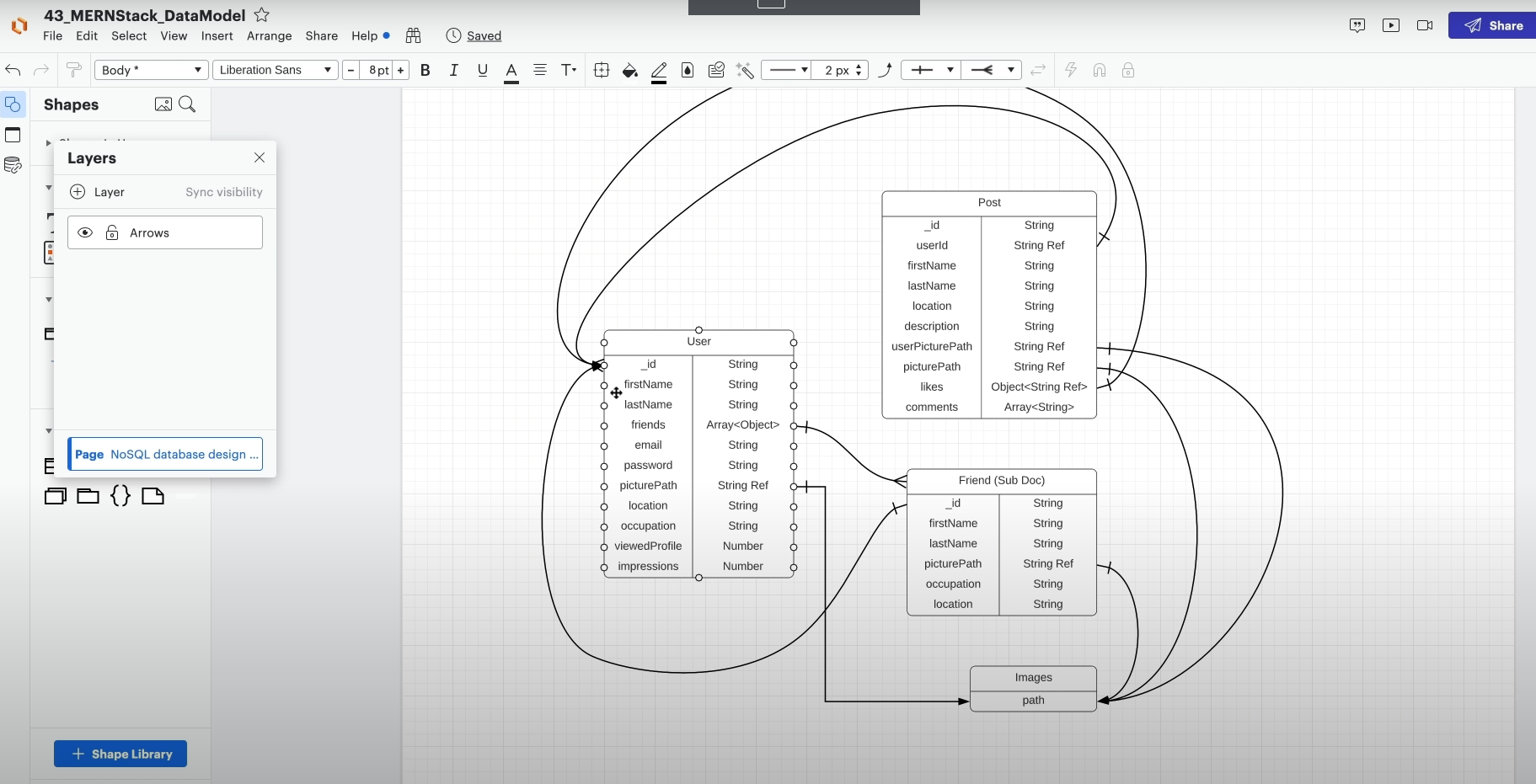
1. Copy the URL and paste it in the .env file and change the password for the same



1. In your code editor write the following code



# Data Design



# Middleware

Middleware are the function which runs between a req and a response eg uploading a picture, verifying the jsonToken

# Steps Followed

Created configurations related to security, storage DB connection

Designing data model

Creating user model (models/User)

Creating endpoint for registration (controllers/auth)

Creating routes

Creating login function

Writing logic to verify jwt token

Creating routes for user data such as getting user data, users friend list and add and remove friends

Creating post routes

# Changes you can make

Uploading the images to the cloud instead of local storage

Make the file uploaded name unique

Check for sttrong authentication process

Implementing sso

Implement proper error handling