PUBLIC GITHUB REPOSITORY URL: <https://github.com/Verdarin/JOSEF_MUSCAT_DGD_6.2A_DE>

PUBLIC API (VERCEL) URL:

TASK 1: SCREENSHOT OF YOUR DEVELOPMENT ENVIRONMENT SETUP:

A screenshot of a computer

AI-generated content may be incorrect.

TASK 2: SCREENSHOT OF YOUR MONGO DATABASE ON MONGOATLAS:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

MONGODB COMPASS SCHEMA:

I used MongoDB’s GridFS this will split the file into smaller chunks and stores them across two collections for audio.chunks and audio.files

Audio.files: will store the metadata of the file : filename, filesize, upload date

Audio.chunks will store the actual file data in chunks

\_id: Auto Generated ObjectID by MongoDB

Data: the binary data of the file chunk itself when combined with “n” field it will become the complete file  
files\_id: reference id to audio.files

n is an integer that is the sequence number of the audio chunk

A screenshot of a computer

AI-generated content may be incorrect.

\_id: Auto Generated ObjectID

Filename: name of the audio file

Length: total size of the audio file in bytes

chunkSize: the size of each chunk GridFS uses 255kb chunks by default

uploadDate: the date and time when the file was uploaded onto the db (autogenerated by GridFS)

A screenshot of a computer

AI-generated content may be incorrect.

Score:

\_id: auto generated ObjectID by MongoDB

Player\_name is a string representing the player’s name

Score is an integer representing the players score

A screenshot of a computer

AI-generated content may be incorrect.

I used MongoDB’s GridFS this will split the file into smaller chunks and stores them across two collections for sprite.chunks and sprite.files

sprite.files: will store the metadata of the file : filename, filesize, upload date

sprite.chunks will store the actual file data in chunks

\_id: Auto Generated ObjectID by MongoDB

Data: the binary data of the file chunk itself when combined with “n” field it will become the image  
files\_id: reference id to sprite.files

n is an integer that is the sequence number of the audio chunk

A screenshot of a computer

AI-generated content may be incorrect.

Sprite.files:

\_id: Auto Generated ObjectID

Filename: name of the audio file

Length: total size of the audio file in bytes

chunkSize: the size of each chunk GridFS uses 255kb chunks by default

uploadDate: the date and time when the file was uploaded onto the db (autogenerated by GridFS)

A screenshot of a computer

AI-generated content may be incorrect.

TASK3A: SCREENSHOT OF THE API RUNNING ON LOCALHOST IN YOUR BROWSER:

A screenshot of a computer

AI-generated content may be incorrect.

TASK3B: SCREENSHOT OF THE HOSTED API RUNNING ON A PUBLIC URL SUCH AS VERCEL

Task 4A: SCREENSHOT SHOWING CREDENTIAL SETUP:

VSCODE:

A screen shot of a computer

AI-generated content may be incorrect.

ATLAS:

A screenshot of a computer

AI-generated content may be incorrect.

PASSWORD: wfBy0qGxscrgcESP

DB NAME: josefmuscat4

TASK4C: SCREENSHOT SHOWING HOW YOU ARE PREVENTING SQL INJECTION.

VSCODE:

A screenshot of a computer

AI-generated content may be incorrect.

THE PlayerScore class im using validation by setting inpuit to be firstly required, minlength of 1 to max length of 50, and a regex pattern for it to follow so they can only input letters and numbers and no special characters

Score is required with it being greater or equal to 0

Sprite.py, Audio.py

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

For Both Audio.py and Sprite.py I am using a Regex to filter out the name of the files

For example, if the filename is "../../etc/passwd", it could be saved in a different directory. The fixed filename would be "\_\_\_\_\_\_etc\_passwd". So we are allowing only letters, numbers, underscore, dash, and dot; replace others with underscore.

I call the method for when I want to CREATE OR UPDATE Sprite / Audio

POSTMAN IDE FOR [POST] PLAYER SCORE:

LOCALHOST API RESPONSE TO [POST]:

MONGO DB VALUES AFTER [POST]:

POST MAN IDE FOR [POST] UPLOAD AUDIO:

LOCALHOST API RESPONSE TO [POST] UPLOAD AUDIO

MONGODB DATABASE AFTER POST:

POSTMAN IDE FOR [POST] UPLOAD SPRITE

API LOCALHOST RESPONSE FOR [POST]:

MONGO DB VALUES AFTER [POST]: