HU3910
Serial Number Database
D22 Group
4/26/2022

Summary of the previous version

The project aims to create a platform for musicians to register and track their instruments. It allows users to create an account with an email address. The main functions of this platform include “Adding” instruments with specific descriptions and tags to the account, claiming the status of instruments (stolen or in possession), “checking” instruments with serial numbers and contacting other users.

E21 group tried to create an image uploading system and you can see an image upload button in “Add”, but it’s not functioning so far.

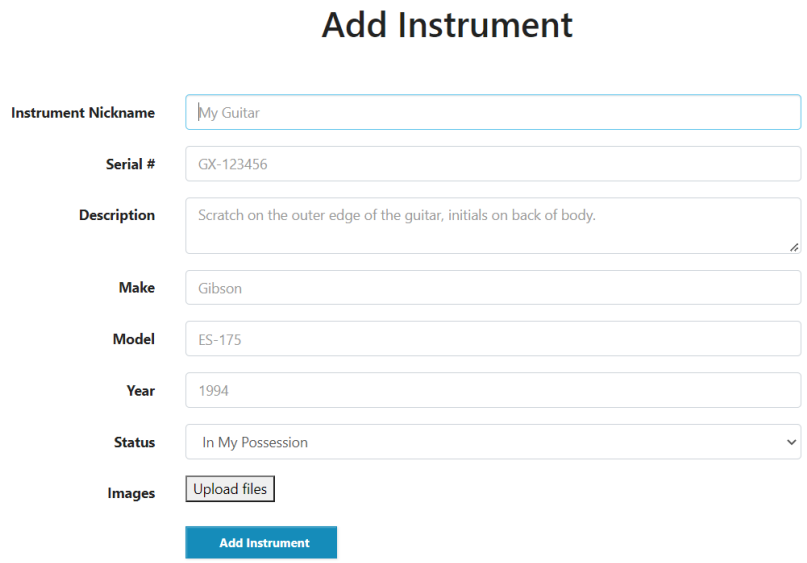


fig.1 Previous “Add”

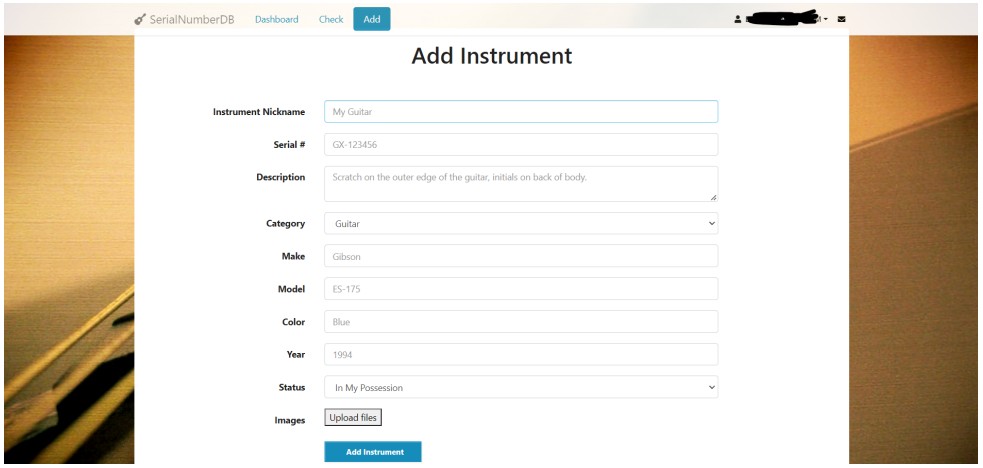


fig.2 updated “Add”

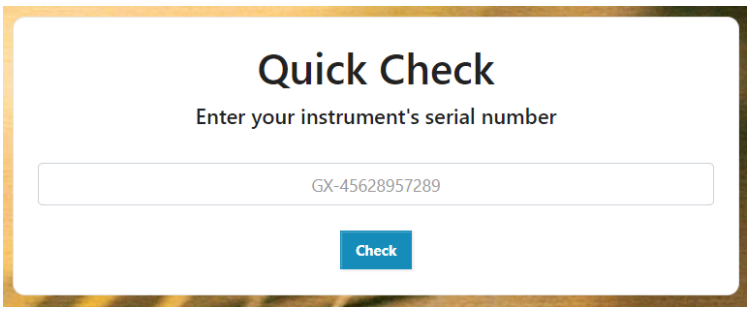


fig.3 Previous “Check”

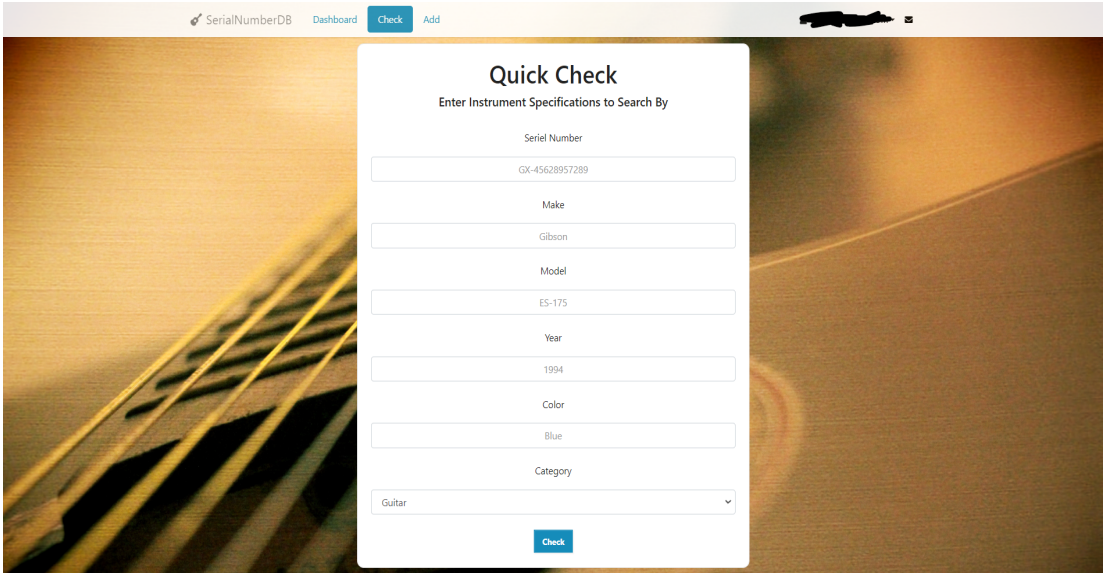


fig.4 Updated “Check”

Updates of the current version

Being able to search not just by serial number but also by categories
Different instruments can have the same serial number - could be dual primary key
Handling duplicate serial numbers.
Added “category” and “color” tags in “Add”.
Added the ability to check by any of the fields that we add to an instrument.

Issues

We decided to focus on front-end developments during this project. Because of this, we have the website looking good for the ADD and CHECK pages, but the info does not pass into the database correctly yet.

Future developments

Utilize Ebay, reverb, craigslist crawlers to get notified when any an instrument with matching criteria gets posted

- Ideally, we can specify the matches that we want to be notified for. Ex: any bass, or just a fender bass

Upload images through cloudinary
Deploy entire database to digital ocean, so it isn’t just on a wpi system

To take over…

1. If https://serialnumberdatabase.wpi.edu/ is not up and running, contact Emerald Toto from IT to start the Docker image that runs the website.

2. Clone the git repository (you will need some proficiency with git)

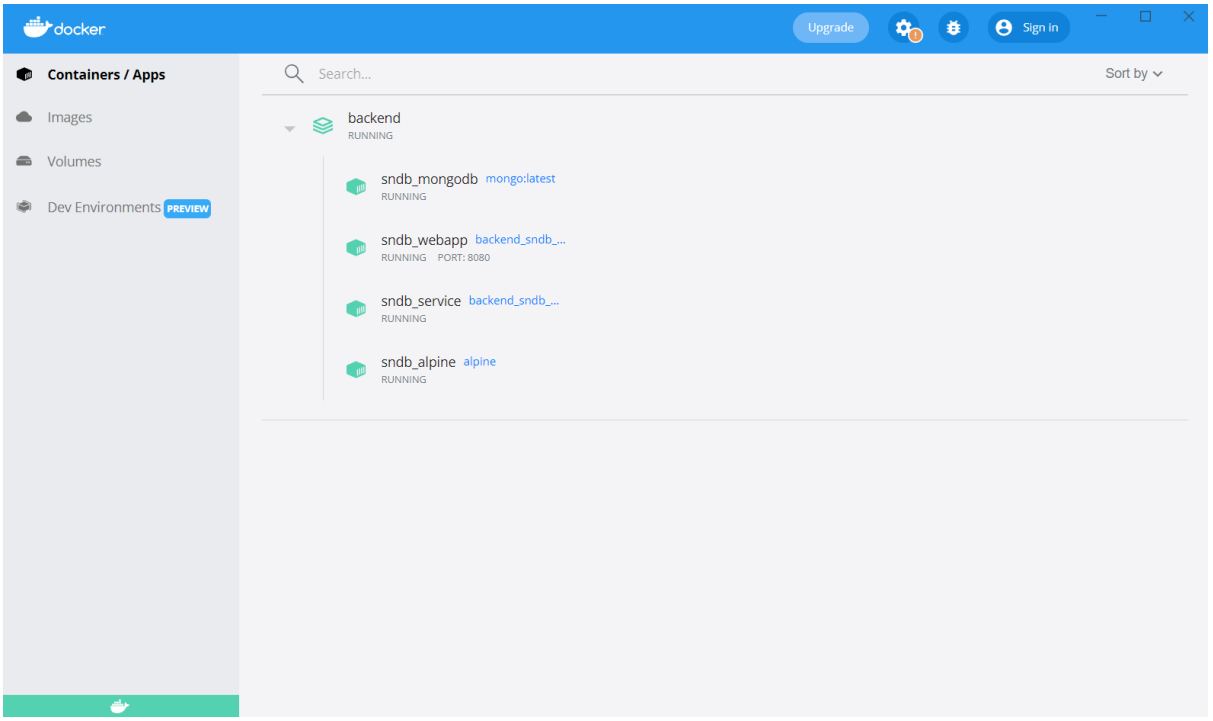
3. Set up the development environment
   It was a bit confusing when our group tried getting started, so here are some tips.
   To get started, go to the Git Repository and read through the readme file.
   You will need to **install Docker and VScode**.
   The latest code is under the ***master*** branch.
   The project keeps being passed from group to group, so some comments might not be specific enough. Check out the Section 0 file structure to get started.

```
. //root folder
├── sndb_mongodb //root folder for mongodb database instance
│   ├── data //database instance
│   │   └── db
│   └── mongodb-database-tools-windows-x86_64-100.0.2
│       └── bin //tools to manage the database
├── sndb_service //root folder for python service
│   ├── sndb
│   │   └── Scripts
│   └── target
├── sndb_webapp //root folder for web application
│   ├── config //list of rules and configurations
│   ├── controllers //list of webapp controllers
│   ├── models //list of models for the mongo db
│   ├── node_modules //npm modules installed for decontainerized work
│   ├── public //public folder visible to the user end
│   │   ├── css
│   │   │   └── themes
│   │   └── js
│   │       └── lib
│   ├── uploads //empty
│   ├── utils //Util taken from stackoverflow answer
│   ├── views //list of pug(another way to write html) files
│   │   ├── account
│   │   └── partials
│   ├── //other notable files lying around
│   ├── .env //needed to config which database you actually connect to
│   └── app.js //the full structure of the express webapp
└── docer-compose.yml //the file you can right click in vscode and compose-up all four containers needed
                      //if you have the docker plugin
```

fig.5 Section 0: File Structure

Use the commands in the ReadMe to:

- Build a Docker Image
- Create the Bridge Network
- Create a new MongoDB container
- Run the `sndb_service:1.0` Docker image in a new container
- Run the web_app either containerized or non containerized



4. Testing
   Test changes by right clicking on the docker-compose.yml file on the left side bar in vs code and click “Compose Up” to run either the containerized or non-containerized docker image depending on how you set up your environment (we used the containerized setup) Then open Docker and you should see the database build.
   - Once you have the docker container running, you can run the launch.json file in vs code. This will run the website on your local host.
     - Be sure to have either line 5 or 6 in the **.env** file uncommented depending on if you are running it containerized or non containerized
   - When you are happy with your changes, push them to the remote origin master branch in git.
   - Once the master branch has your changes, https://serialnumberdatabase.wpi.edu/ should reflect them as well.