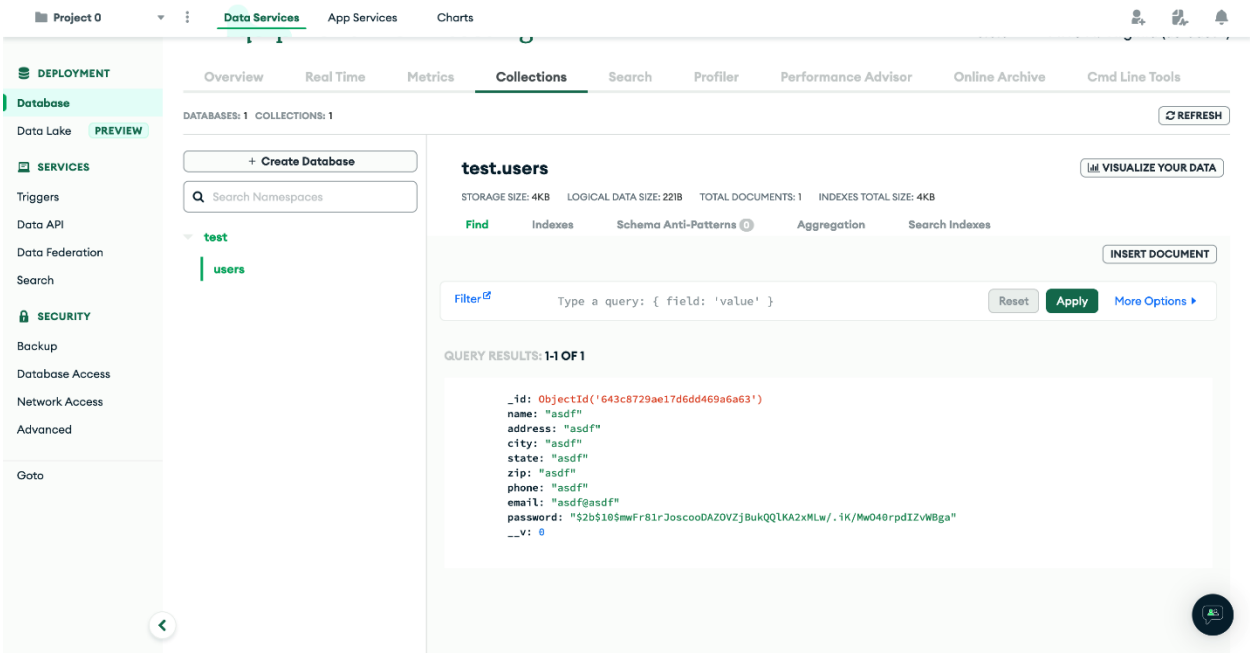
Part 5 : Database Design Document

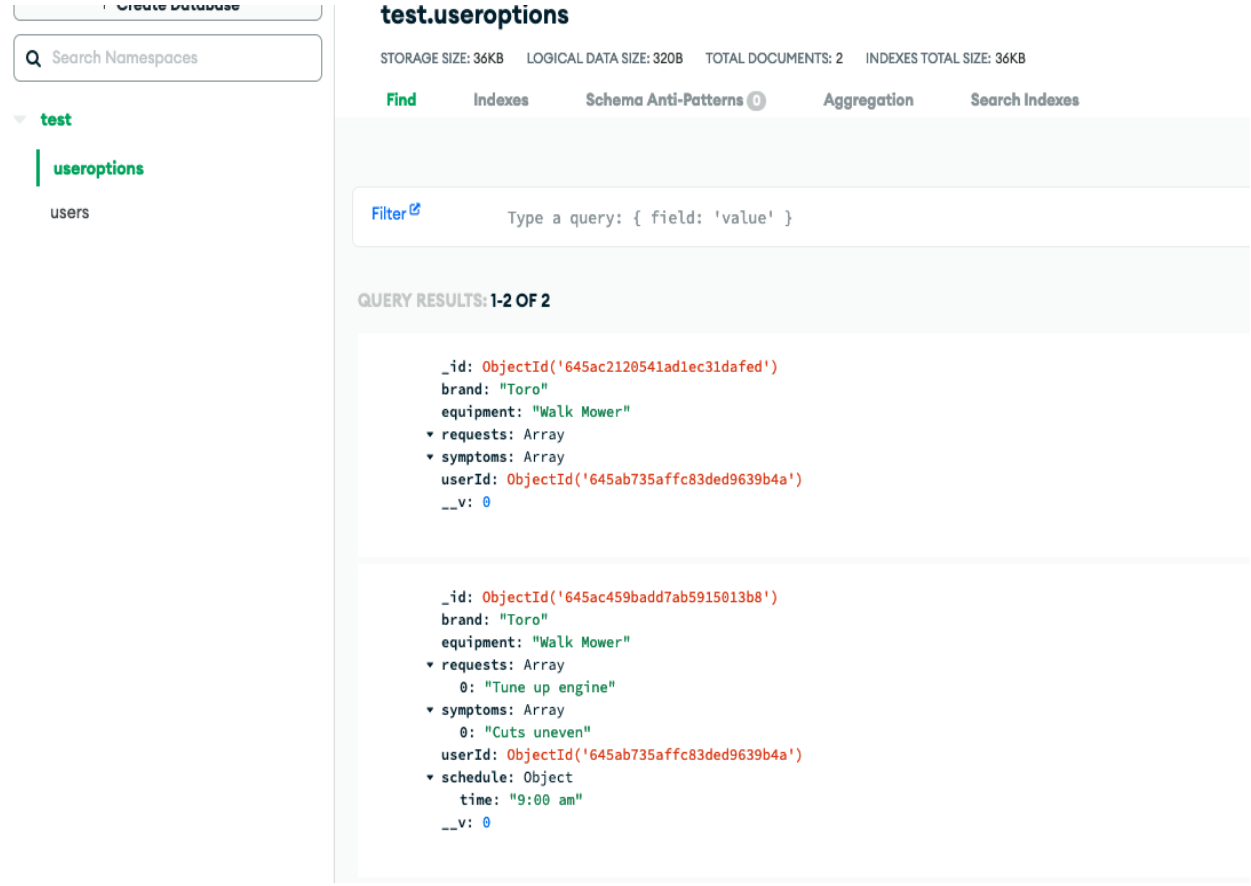
The first schema used will be that of a user .



This is an example of a user with the fields name , address , city , state , zip , phone , email , and password.

```
// Create a new user with the hashed password
const newUser = new User({
    name: req.body.name,
    address: req.body.address,
    city: req.body.city,
    state: req.body.state,
    zip: req.body.zip,
    phone: req.body.phone,
    email: req.body.email,
    password: hashedPassword,
});
```

The second schema used will collect the data of an existing user.




```
if (req.session && req.session.user) {
    const userOptions = new UserOptions({
        brand,
        equipment,
        requests,
        symptoms,
        schedule,
        userId: req.session.user._id,
    });
```

The type of equipment and brand are chosen along with an array that includes the symptoms and request being made.

Added to this will be the date and time chosen by the client and incorporated into this data set.

```
if (req.session && req.session.user) {
    const userSchedule = new Schedule({
        date,
        time,
        description,
        userId: req.session.user._id,
    });
```

Initially an existing user may attempt to log in. Commands such as findOne() will be needed to determine if an existing user exists.

```
app.post("/login", async (req, res) => {
    // Find the user in the database using the provided email
    const user = await User.findOne({ email: req.body.email });
```

Potentially options can be added allowing users to edit their information .

MongoDB stands as the only document database in use. Very flexible and scalable as additional features can be added and are being added as this document is being produced.