Matt Burton

Milestone 2

Framework Stacks:

‘MEAN’ stack:

Mongo DB
ExpressJS
AngularJS
NodeJS

‘FARM’ stack:

FastAPI
ReactJS
Mongo DB

Web Design Frameworks

Common Framework features:

1. Routing: allows developers to map URLs to application code
2. Templates: provide a way to separate presentation logic from business logic
3. Object-Relational Mapping (ORM): simplify database interactions
4. Authentication and Authorisation: provide security mechanisms to protect web applications
5. Caching: improve application performance by storing frequently accessed data in memory
6. Testing: support automated testing of application code

I used the FARM (FastAPI, React, MongoDB) stack for this website development project. This stack combines the FastAPI Python framework for building APIs, the React JavaScript library for building user interfaces, and a MongoDB database for data storage.

The purpose of Web frameworks:

Web design frameworks are intended to aid developers in constructing websites. Pre-written code, tools, and resources are gathered together to help build websites that are attractive, accessible, and responsive.

The reason for using a web design framework is so that developers may save time and effort. Developers can then avoid writing every line of code from the ground up by using pre-built components and templates.

Web design frameworks allow us to build websites using the most recent methods. They are often updated regularly by a group of developers that work to find and fix bugs and vulnerabilities as well as offer new functionality and features.

Features to be used:

FARM Stack:

- FastAPI for building APIs
    - Routing
    - Object-Relational Mapping (ORM)
    - Authentication and Authorization
    - Testing
- React for building user interfaces
    - Templates
    - User Interface components
- MongoDB for data storage
    - Object-Relational Mapping (ORM)
    - Caching

MEAN Stack:

- MongoDB for data storage
    - Object-Relational Mapping (ORM)
    - Caching
- Express for building APIs
    - Routing
    - Object-Relational Mapping (ORM)

    - Authentication and Authorization
    - Testing
- Angular for building user interfaces
    - Templates
    - User Interface components

Emerging Website Technology

Chosen technology:

Immersive View is an emerging web technology that enables website visitors to experience 360-degree panoramic views of physical spaces.
It is built on Google Maps and uses high-resolution imagery and computer vision to create an immersive and interactive user experience.
This emerging web technology links to my plan for a website as it would give users an improved ability to find and understand where they need to go to find the food that is being advertised and may be received.
Unfortunately, this technology hasn't been released just yet, therefore I will be focusing on the maps functionality which is constantly growing and updating to the changing world, in particular the ability to focus in on one area and provide markers for the places that the food is to be picked up from, by the beneficiary once the charity member has accepted a request to pick up the advertised excess food, although this emerging technology is not yet released it is growing and relevant to the design.

How to implement the technology:

Create a Google Maps API Key: To use Immersive View, I first create a Google Maps API key. This will allow the website to communicate with Google Maps.

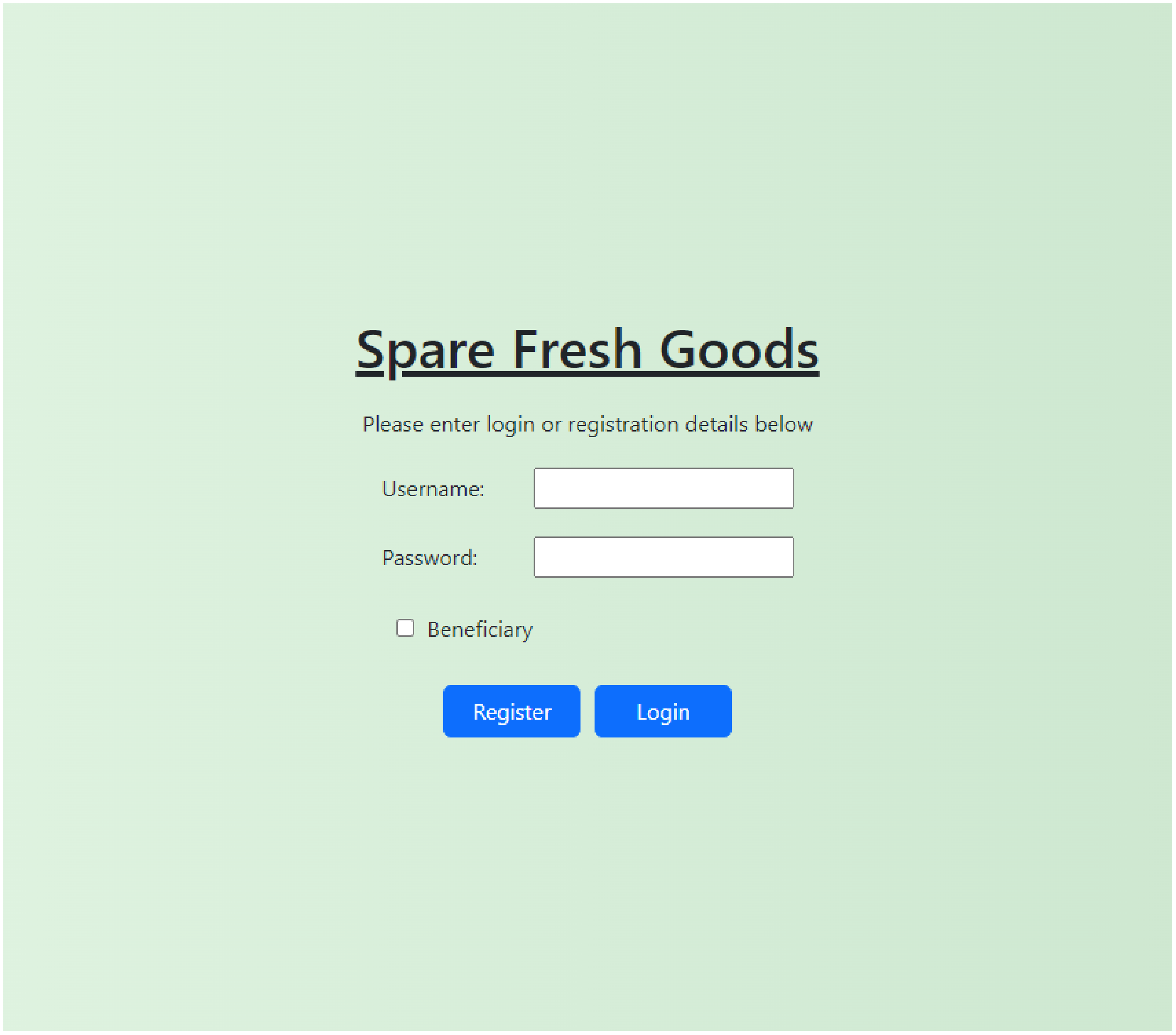
Integrate the Google Maps API: Once I have a Google Maps API key, I integrate the Google Maps API into my website's code. Then use the Google Maps JavaScript API to add the map functionality to the website.

The code for incorporating the maps on the frontend:

```
@app.get("/", response_class=HTMLResponse)
async def get_map():
    return """
    <html>
    <head>
        <script
src="/static/js?key=<API_KEY_HERE>&libraries=places"></script>
        <style>
            #map {
                height: 400px;
                width: 100%;
            }
        </style>
    </head>
    <body>
        <div id="map"></div>
        <script>
            function initMap() {
                const map = new
google.maps.Map(document.getElementById("map"), {
                    center: { lat: -34.397, lng: 150.644 },
                    zoom: 8,
                });
            }
        </script>
        <script
src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=init
Map"
            async defer></script>
    </body>
    </html>
    """
```

FARM stack:

Login/registration page:



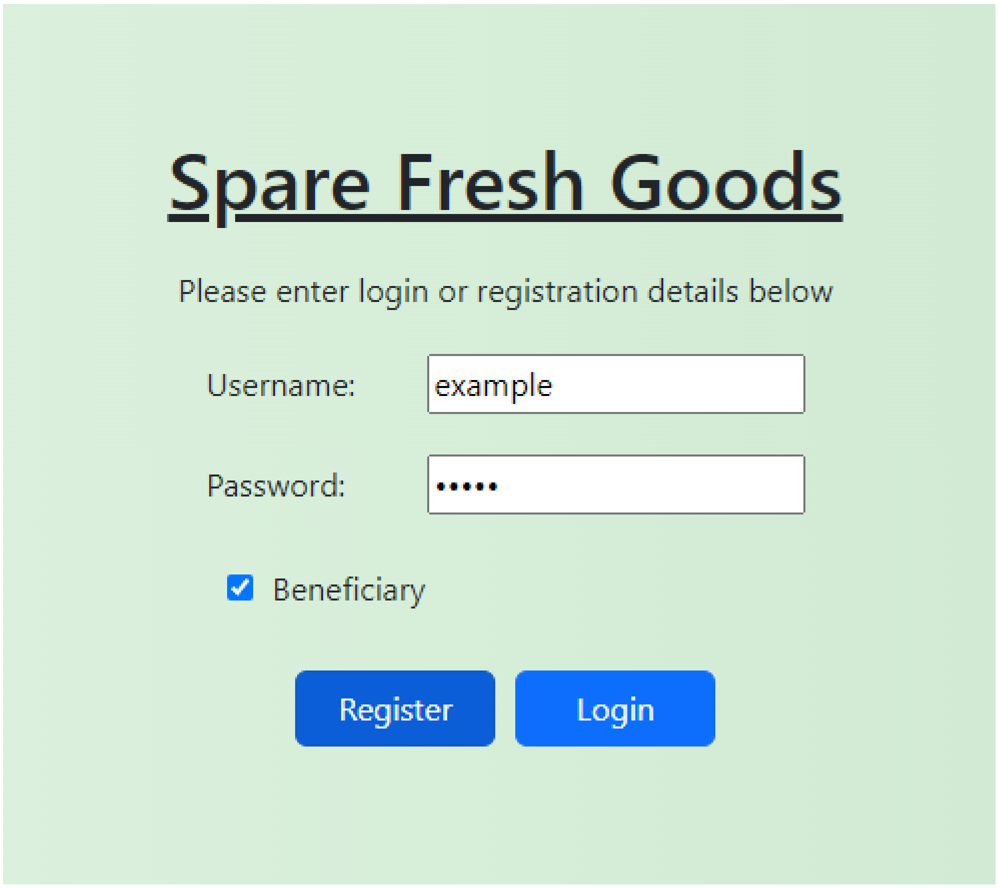
Spare Fresh Goods

Please enter login or registration details below

Username:

Password:

Example details:



Spare Fresh Goods

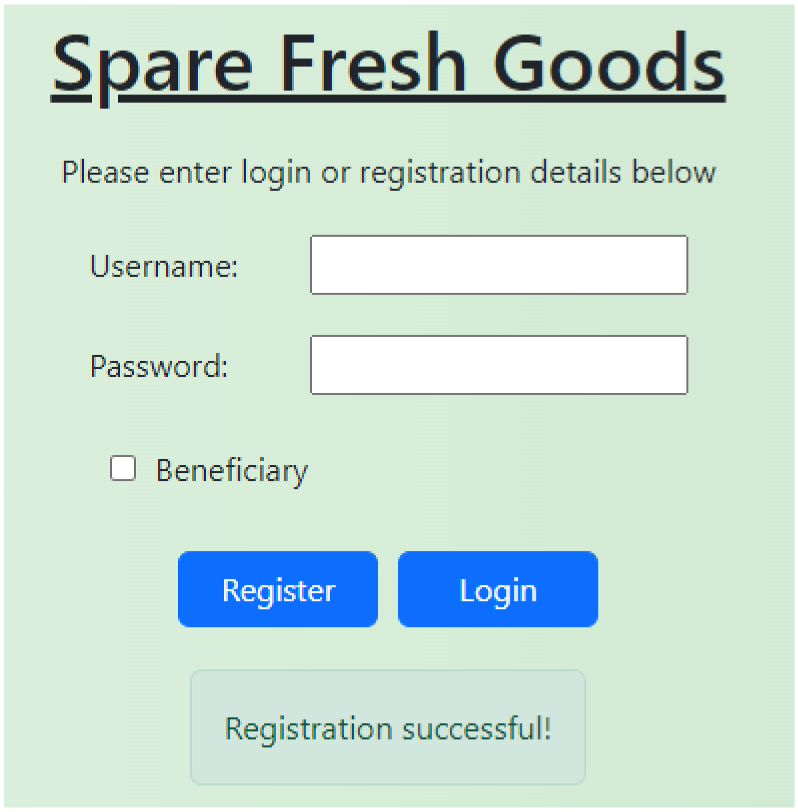
Please enter login or registration details below

Username: example

Password: •••••

[✓] Beneficiary

Registration success message:



Spare Fresh Goods

Please enter login or registration details below

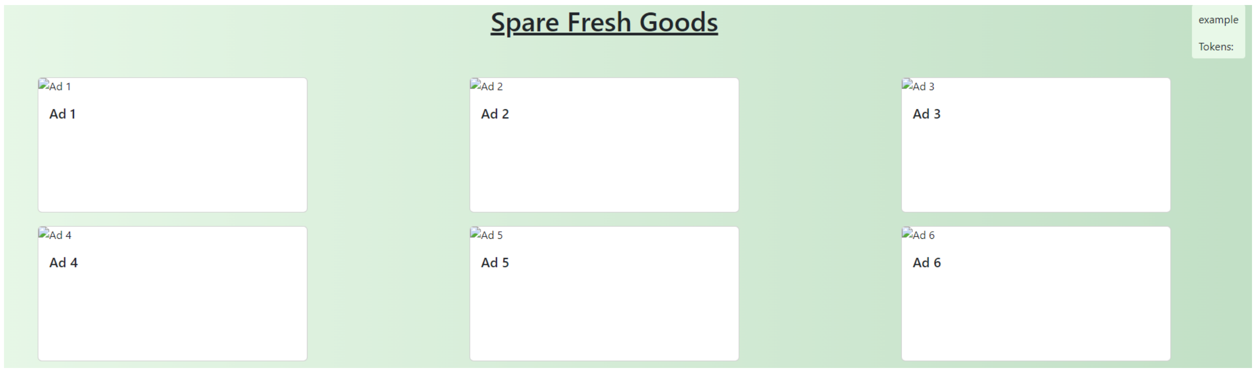
Username:

Password:

☐ Beneficiary

Registration successful!

Home page:

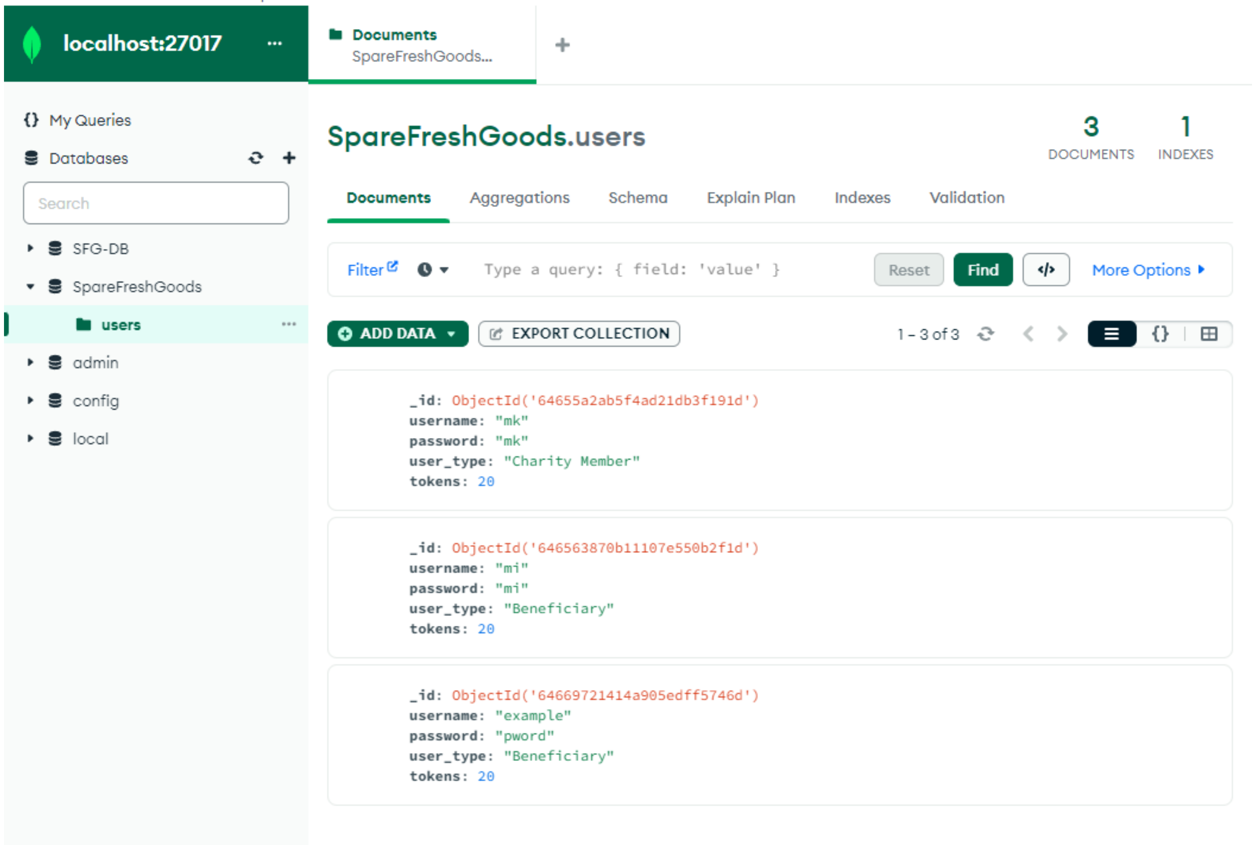


Spare Fresh Goods

example

Tokens:

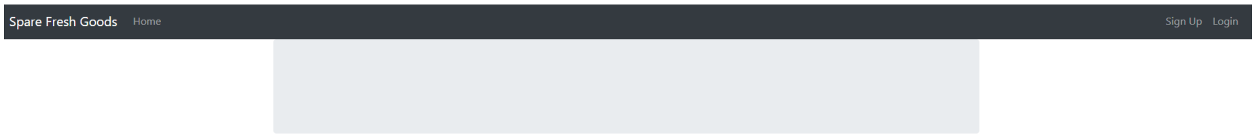
Database storage:



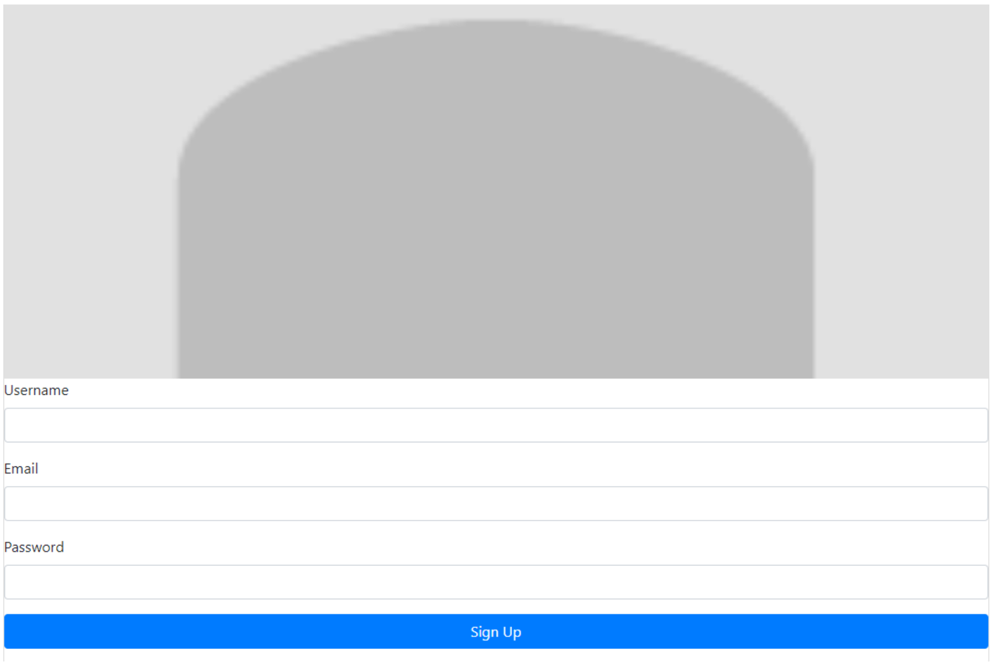
```
_id: ObjectId('64655a2ab5f4ad21db3f191d')
username: "mk"
password: "mk"
user_type: "Charity Member"
tokens: 20

_id: ObjectId('646563870b11107e550b2f1d')
username: "mi"
password: "mi"
user_type: "Beneficiary"
tokens: 20

_id: ObjectId('64669721414a905edff5746d')
username: "example"
password: "pword"
user_type: "Beneficiary"
tokens: 20
```

MEAN stack:

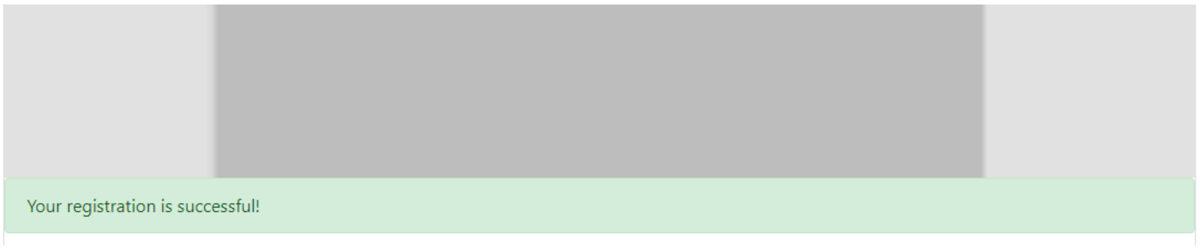
Landing page:



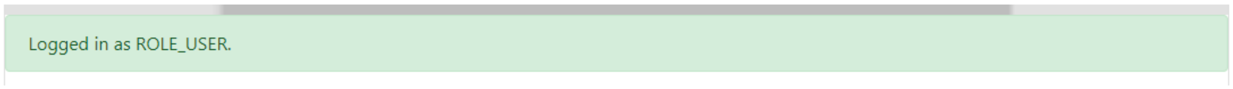
Registration page:



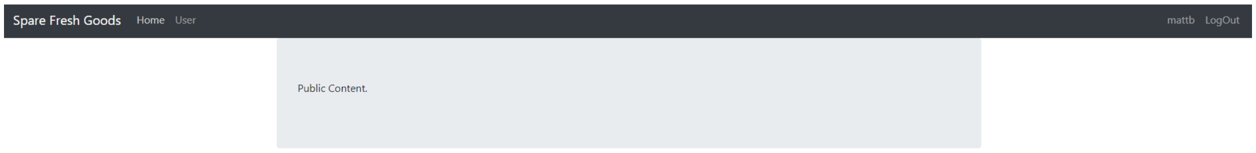
Registration success message:


Your registration is successful!

Login success message:

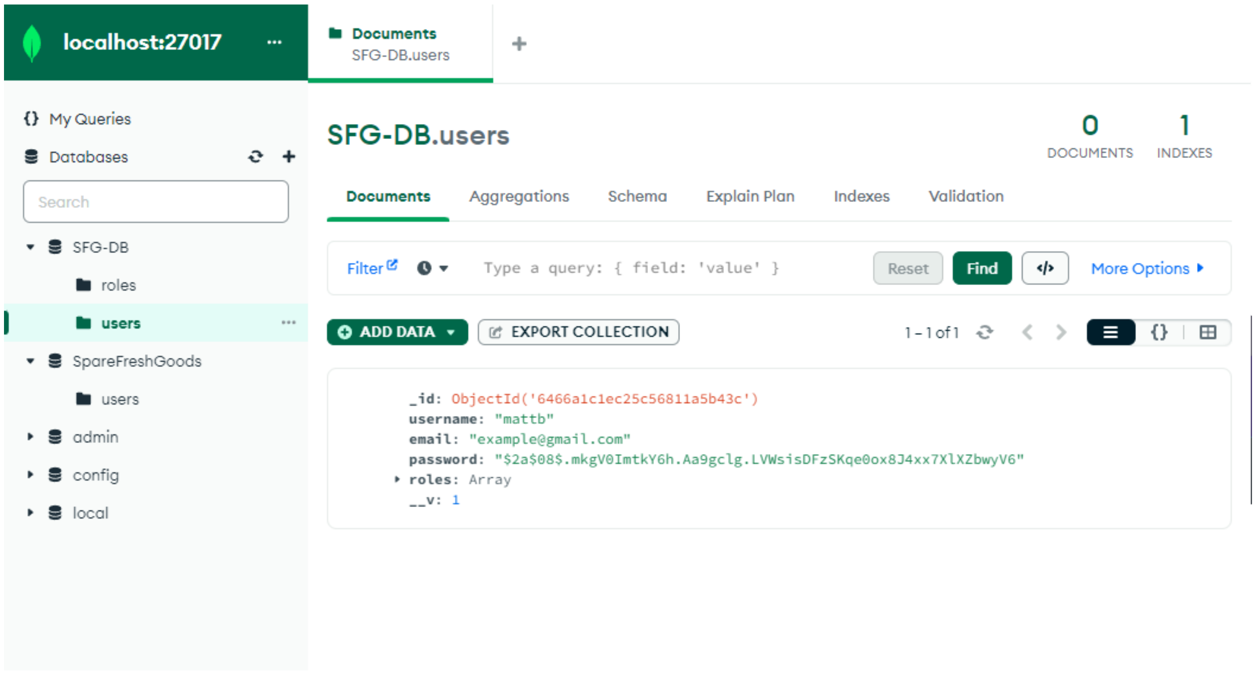

Logged in as ROLE_USER.

User home page:



Public Content.

Database storage:



```
_id: ObjectId('6466a1c1ec25c56811a5b43c')
username: "mattb"
email: "example@gmail.com"
password: "$2a$08$.mkgV0ImtkY6h.Aa9gclg.LVWsisDFzSKqe0ox8J4xx7XlXZbwyV6"
roles: Array
__v: 1
```

Stack recommendation:

FARM Stack (FastAPI, React, MongoDB):

Pros:

- FastAPI: FastAPI is a modern, fast, and highly efficient Python web framework. It provides excellent performance and scalability.
- React: React is a popular JavaScript library for building user interfaces. It allows for efficient rendering and component-based development, resulting in a smooth and interactive user experience.
- MongoDB: MongoDB is a NoSQL database that offers flexibility and scalability. It is easy to work with and can handle large amounts of data.

Cons:

- Learning Curve: If you are new to Python or React, there may be a learning curve involved in getting up to speed with these technologies.
- Limited Server-Side Rendering: FastAPI is primarily focused on building APIs, so server-side rendering is not as straightforward as with other frameworks.

MEAN Stack (MongoDB, Express.js, Angular, Node.js):

Pros:

- Widely Used: The MEAN stack is widely adopted and has a large community of developers, which means there are abundant resources and support available.
- Full-Stack JavaScript: With MEAN, you can use JavaScript throughout the entire development stack, which can simplify the development process if you are already familiar with JavaScript.
- Server-Side Rendering: Angular, one of the components of the MEAN stack, supports server-side rendering out of the box, which can improve initial page load times and SEO.

Cons:

- Complexity: The MEAN stack is more complex compared to the FARM stack, especially if you are new to the technologies involved. It requires a solid understanding of JavaScript and can have a steeper learning curve.
- Performance Overhead: The full-stack JavaScript approach may introduce some performance overhead compared to more lightweight frameworks.
- Larger File Sizes: Angular, in particular, can result in larger file sizes, which may impact the initial load time of the website.

Considering the factors mentioned and the pros and cons of each stack, the FARM stack appears to be a suitable choice for my website. It offers ease of testing, faster development, clear implementation, and smaller file sizes.