Crypto Keys Storage

Design and implement an application that will securely store secrets (cryptographic keys and passwords). The application will allow users to authenticate and securely add/generate keys and passwords and keep them safe, revealing them only to their respective owners.

The application will minimally consist of:

- A web or desktop-based interface in which the users will manage their private data.
- A database, which will store cryptographic keys and secrets.
- A server component.

Presenting the application:

- The code must be available on a public git repository (e.g., github).
- Each team must make:
    - A max 5-minute presentation about design and implementation of each given requirement.
    - A max 5-minute demo (e.g., Adding 5 users, encrypt & decrypt a given password).
- A max 5-minute Q&A session will follow after each presentation.

Requirements:

- A user can create an account and log in.
- Users can add/delete their own cryptographic keys or passwords.
- The keys and passwords can only be viewed and managed by their respective owners (no other user, including the admin, can view the secrets).
- All data transfers must be done securely (between all application layers).
- Users can generate keys within the application and then store them if they want.
    - Types of keys that must be generated and stored:
        - Symmetric Keys:
            - AES (128, 192 or 256 bits)
            - Triple DES

        - Asymmetric Keys:
            - RSA
            - ECC (Elliptic Curve Cryptography)
- Users can generate passwords for specific usernames within the application and store them.
    - Password generation is based on some user-configurable parameters (e.g., length, password structure)
- The database must be separated from the backend (In a separate container/VM where exposed services should be limited)
- Clean and secure coding.

Some questions to think about when designing the application:

- How secure is account management?
- How easily scalable is the application?
- What if the database gets leaked?

Notes:

- There are no restrictions regarding used technologies and programming languages. You are free to use what suits you best.
- You don't need to implement all functionalities. Each one will be individually graded.