Regarding the log you've provided:

- **`docker-compose run --rm backend flask db init` output:**
  - The "WARN" message about the `version` attribute being obsolete is normal; it's a general warning from Docker Compose that you can ignore for now.
  - The lines indicating "Creating directory /app/migrations ... done", "Creating directory /app/migrations/versions ... done", and "Generating /app/migrations/alembic.ini ... done", etc., **are all normal and expected** when you run `flask db init` after deleting the `migrations` directory. This confirms that Alembic has successfully re-initialized its directory structure.
- **`docker-compose run --rm backend flask db migrate -m "Initial schema setup with all models and fields"` output:**
  - The "WARN" and "INFO" messages are normal.
  - However, the `ERROR [flask_migrate] Error: Can't locate revision identified by '93cdd7320c88'` is **not normal** in this context. This error means Alembic is still looking for a previous migration record that doesn't exist in its newly initialized state.
- **`docker-compose run --rm backend flask db upgrade` output:**
  - Similarly, the `ERROR [flask_migrate] Error: Can't locate revision identified by '93cdd7320c88'` here confirms the same underlying issue.

The Problem:

The persistence of the `Can't locate revision identified by '93cdd7320c88'` error after performing `rm -r backend/migrations/` and `flask db init` suggests one of two possibilities:

1. **The `backend/migrations/` directory was not fully or correctly deleted by `rm -r backend/migrations/` before `flask db init` was run.** This could happen if there were permissions issues, or if the `rm -r` command was not executed correctly or from the right directory on your host machine. If old files or a `.git` history of them persist, `flask db init` might still see some old state.
2. **There's an issue with the Docker volume mapping or caching that is preventing the host `migrations` directory from being truly fresh inside the container.** This is less common but can occur.

What to do next:

Let's try to ensure a *completely clean slate* for your `migrations` directory and database:

1. **Stop and remove everything related to your Docker Compose project:**
   Bash

```
docker-compose down --volumes --rmi all
```

2.

    - `--volumes`: This is crucial. It will remove the `network_doc_db` named volume, effectively deleting your SQLite database file and its internal Alembic history.
    - `--rmi all`: This will remove all images created by `docker-compose`, ensuring a fresh build.

3. **Manually verify and clean the `backend/migrations/` directory on your host machine:**
   Bash

```
# From your network-doc-app root directory
```

4. `rm -rf backend/migrations/`
5.

After running this, confirm that the `backend/migrations/` directory is completely gone from your file system. If it's not, try deleting it manually through your file explorer.

6. **Now, run the Flask-Migrate commands again, step-by-step:**
    - **Initialize Flask-Migrate from scratch:**
      Bash

```
docker-compose run --rm backend flask db init
```

    -

This *must* create the `migrations` directory cleanly.

- **Create a new migration script:**
  Bash

```
docker-compose run --rm backend flask db migrate -m "Initial schema setup with all models and fields"
```

This should now generate a single, fresh migration file in `backend/migrations/versions/`.

- **Apply the new migration:**
  Bash

```
docker-compose run --rm backend flask db upgrade
```

This should execute successfully without errors.

7. **Finally, start your full application:**
   Bash

```
docker-compose up -d --build
```

8.

This process should thoroughly clean up any lingering migration history or database state, allowing Alembic to start fresh.