

1. Planning our models

Our database will consist of 2 different collections:

- A User collection, that has an email, first name, last name, a hashed password, and a list of complaints references(ids) and a role, picked from a set of enum (vip, regular, admin).
- A Complaint collection, that has a title, a body, a status enum that defaults to 'PENDING', a creator referencing to the user, and createdAt, updatedAt timestamps.

Upon signup if no query parameter was passed, the signed up user will default to a regular(non-Vip) user,

2. API Workflow

The following explains how the API is expected to be used:

- First a user must sign up with his email address , first name, last name, and a password along with the vip or admin option, no user can have multiple roles at the same time.
- If the user is either Vip or Regular, they can create new complaints, see their own complaints, or see a single complaint.
- If the user is an Admin, he can update a complaint status by status id using url parameters
- The admin can see all complaints either only sorted by most recent, or categorized by status and sorted by most recent.

3. Additional info

MongoDB Compass url: `mongodb+srv://josepham:josepham@nestjs.mzhpmai.mongodb.net/test`

Sample users:

- | | |
|---|------------------------|
| • Email: userRegular1@mail.com | password: userRegular1 |
| • Email: userRegular2@mail.com | password: userRegular2 |
| • Email: userVip1@mail.com | password: userVip1 |
| • Email: userVip2@mail.com | password: userVip2 |
| • Email: userAdmin@mail.com | password: userAdmin |

For ease of use, a get to <http://localhost:3000/complaints/admin/id> request was made to fetch all complaints with their ids for faster testing (updating statuses etc..)