HOUSEHOLD SERVICES APPLICATION - V2

PROJECT REPORT

Prepared By:
ARYAN KOHLI

Roll no:

22f3001832

PROJECT DETAILS:

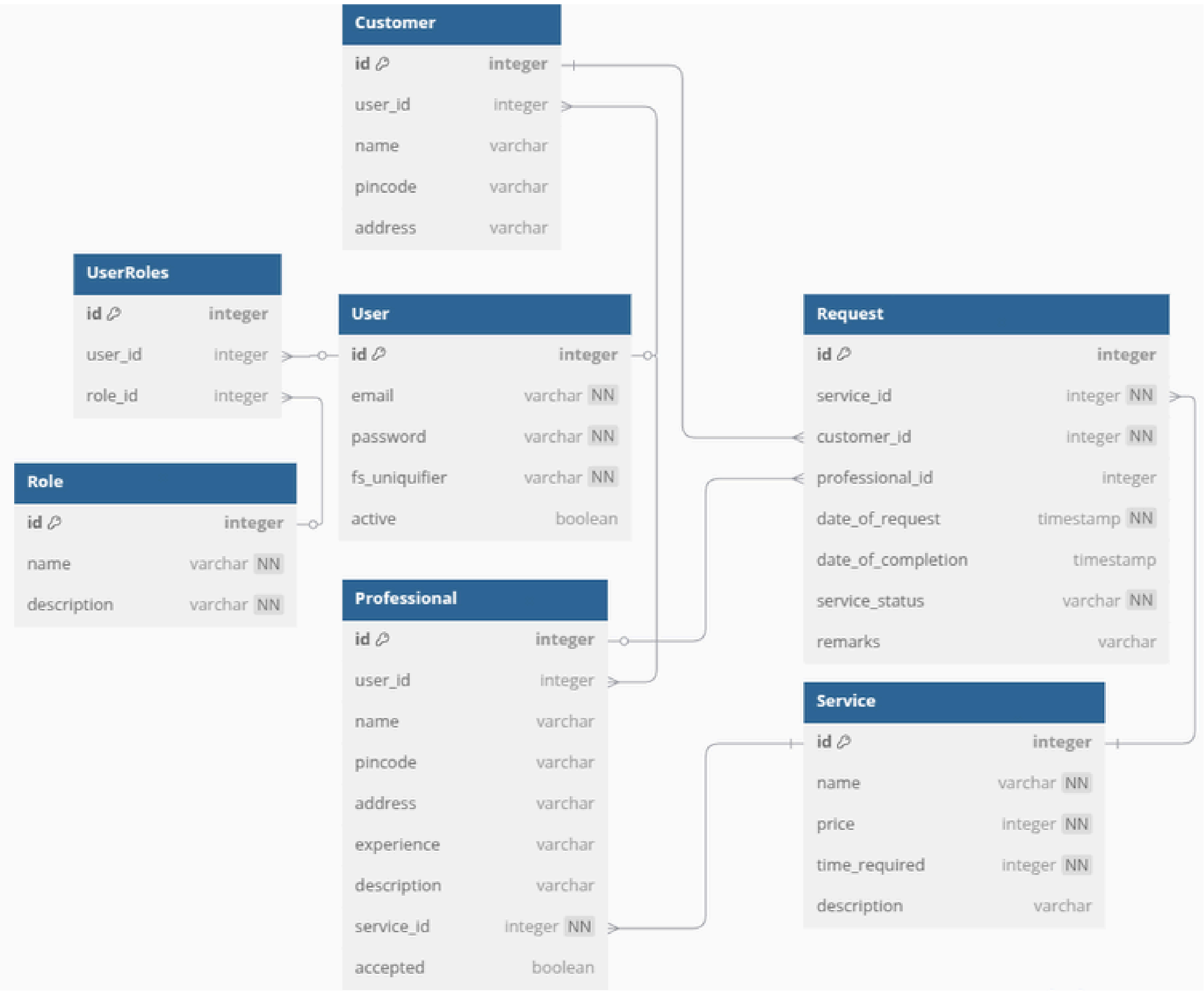
Problem Statement:

To make a multi-user app which acts as platform for providing comprehensive home servicing and solutions. Customers can book services they need such as plumbing, lawn care, laundry service, etc... whereas professionals can register them on app to provide their service. Once they are verified by admin, they start receiving service requests.

Technologies Used:

1. **Flask:** A lightweight Python framework used for backend API development.
2. **VueJS:** A JavaScript framework designed for building dynamic and interactive user interfaces.
3. **Bootstrap:** A CSS framework that ensures responsive and visually appealing designs.
4. **Flask-SQLAlchemy:** An ORM tool for simplifying database management and interactions.
5. **Flask-Security:** Simplifies adding authentication, role management, and security features to Flask apps.
6. **Flask-Caching:** Improves application performance with caching capabilities.
7. **Celery:** Handles background tasks like reminders and periodic reports asynchronously.
8. **SQLite:** A lightweight database solution for storing user data and service-related information.

DB Schema:



Customer	
id	integer
user_id	integer
name	varchar
pincode	varchar
address	varchar

UserRoles	
id	integer
user_id	integer
role_id	integer

User	
id	integer
email	varchar NN
password	varchar NN
fs_uniquifier	varchar NN
active	boolean

Role	
id	integer
name	varchar NN
description	varchar NN

Professional	
id	integer
user_id	integer
name	varchar
pincode	varchar
address	varchar
experience	varchar
description	varchar
service_id	integer NN
accepted	boolean

Request	
id	integer
service_id	integer NN
customer_id	integer NN
professional_id	integer
date_of_request	timestamp NN
date_of_completion	timestamp
service_status	varchar NN
remarks	varchar

Service	
id	integer
name	varchar NN
price	integer NN
time_required	integer NN
description	varchar

ER Diagram of Application

Model Used:

1. **User:** Used to keep track of all users.
2. **Role:** Customer, Professional, Admin
3. **UserRoles:** Role of Users
4. **Customer:** Customer Details
5. **Professional:** Professional Details
6. **Services:** Used to keep track of Services
7. **Request:** Request Details

Directory Structure:

```
MAD_2_Project/
├─── app.py
├─── celery_worker.py
├─── frontend/
│    ├─── app.js
│    ├─── index.html
│    ├─── components/
│    │    ├─── Navbar.js
│    │    └─── ServicsCard.js
│    ├─── pages/
│    │    ├─── admin/
│    │    │    ├─── AdminDashboard.js
│    │    │    ├─── EditServiceDisplay.js
│    │    │    ├─── NewService.js
│    │    │    ├─── Block.js
│    │    │    ├─── EditService.js
│    │    │    └─── ProfManagement.js
│    │    ├─── customer/
│    │    │    ├─── CompleteProfileCustomer.js
│    │    │    ├─── CustomerRequest.js
│    │    │    ├─── ProfileEdit.js
│    │    │    └─── Review.js
│    │    ├─── prof/
│    │    │    ├─── CompleteProfileProf.js
│    │    │    ├─── ProfDashbord.js
│    │    │    └─── ReqProfessional.js
│    │    ├─── LoginPage.js
│    │    ├─── ProfessionalReview.js
│    │    ├─── RegisterPage.js
│    │    ├─── ServiceDisplay.js
│    │    └─── SrevicesList.js
│    └─── utils/
│       ├─── router.js
│       └─── store.js
├─── backend/
│    ├─── celery/
│       ├─── celery_aryan.py
│       ├─── celery_schedule.py
│       ├─── mail_service.py
│       ├─── tasks.py
├─── config.py
├─── create_initial_data.py
├─── instance/
├─── models.py
├─── resources.py
├─── routes.py
```

Important API’s Used:

- Authorization:
  - Register: /register
  - Log in: /login

- Services:
  - Services: /api/services
  - Particular Service: /api/services/<int:service_id>

- Request:
  - Request: /api/request

- Professional:
  - Professional: /api/prof
  - Particular Professional: /api /prof/<int:service_id>

- Customer:
  - Customer: /api/cust

- Review:
  - Reviews: /api//review/<int:request_id>

Demonstration Video:
https://drive.google.com/file/d/10JmOcXPuecY45RMZ6blqUNcxcC2DXJvZ/view?usp=sharing