Project Report

Household Services

Student Details

- **Name** : Ripunjay Kumar
- **Roll No** : 21f3002511@ds.study.iitm.ac.in
- **Project Title** : Household Services

Project Details

- **Question Statement** : Design and implement a comprehensive household services platform that connects customers with service professionals under administrative oversight. The platform should facilitate the booking, management, and completion of household services while ensuring quality control through user reviews and administrative monitoring.

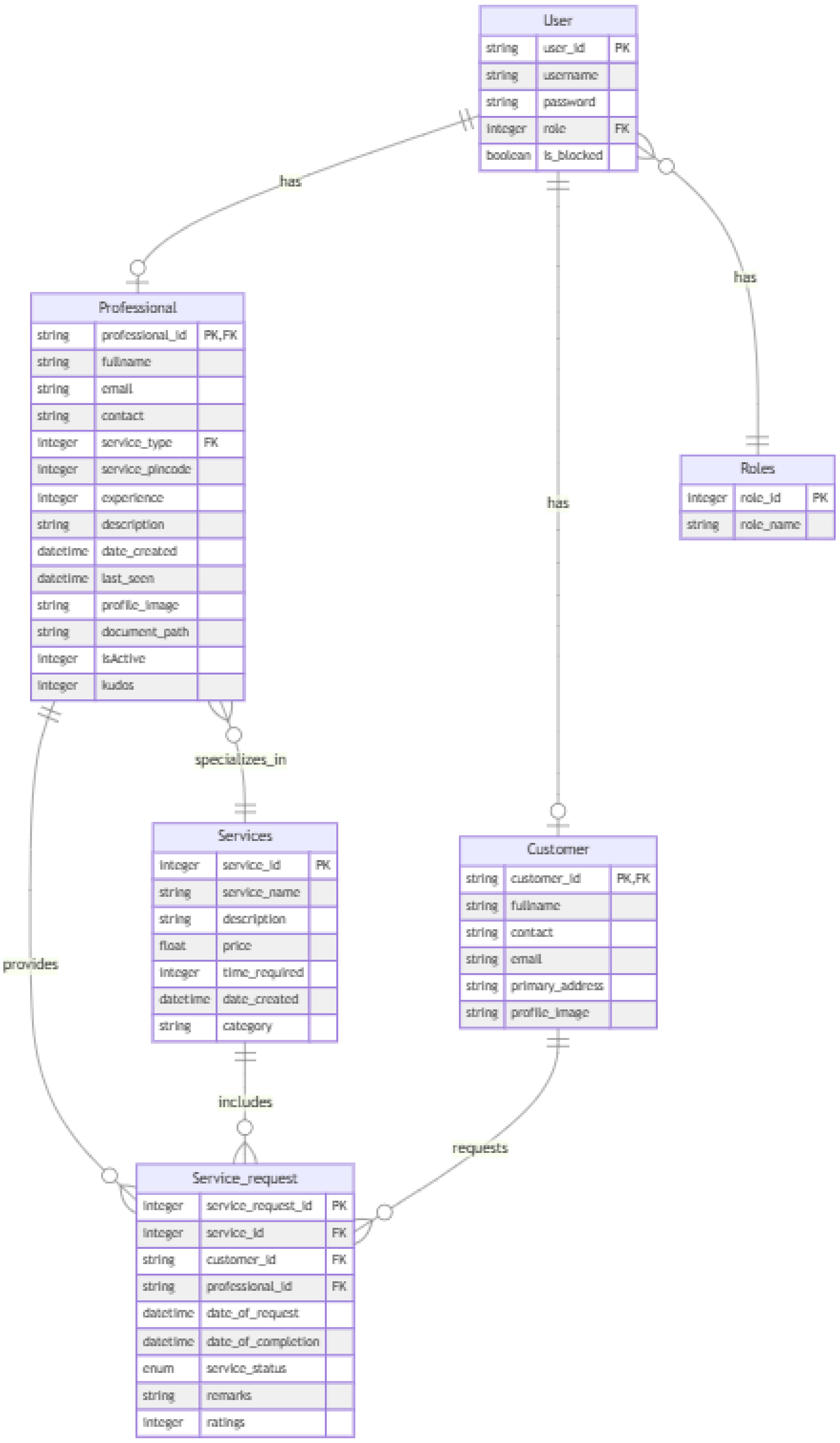
- **Approach to Problem Statement**: I first established the database schema and relationships between users, services, and service requests using SQLite, where I implemented proper indexing and constraints for data integrity. I then developed the Flask backend with clear API endpoints for authentication, service management, and request handling, implementing JWT tokens for secure role-based access. Next, I created background job processors using Redis and Celery that efficiently handled notifications, report generation, and CSV exports asynchronously to prevent blocking operations. For the frontend, I built reusable Vue components with Bootstrap styling, starting with authentication flows and then developing role-specific dashboards. I implemented Redis caching for frequently accessed data and optimized API performance through efficient database queries. Additionally, I integrated a sentiment analysis model that analyzes customer reviews and feedback in real-time, providing valuable insights to service professionals and administrators about service quality and customer satisfaction trends..

Frameworks & Libraries Used

- **Backend Framework and APIs**
  - Flask-RESTful used for creating RESTful API endpoints
  - Implemented resource-based routing
- **Authentication & Authorization**
  - JWT (JSON Web Tokens)Implemented secure user authentication
  - Role-based access control (Admin, Service Professional, Customer)
  - Token-based session management
- **Frontend**
  - Vue.js Component-based UI development
  - Route handling
  - Form validations
  - UI components
  - Styling utilities
- **Caching and Background Jobs**
  - Redis for data caching
  - Managed background job queues
  - Stored session information

- **Database**
  - SQLite - Main data storage
  - Relationship management
  - Query optimization

ER-Diagram



User		
string	user_id	PK
string	username	
string	password	
integer	role	FK
boolean	is_blocked	

Professional		
string	professional_id	PK,FK
string	fullname	
string	email	
string	contact	
integer	service_type	FK
integer	service_pincode	
integer	experience	
string	description	
datetime	date_created	
datetime	last_seen	
string	profile_image	
string	document_path	
integer	isActive	
integer	kudos	

Roles		
integer	role_id	PK
string	role_name	

Services		
integer	service_id	PK
string	service_name	
string	description	
float	price	
integer	time_required	
datetime	date_created	
string	category	

Customer		
string	customer_id	PK,FK
string	fullname	
string	contact	
string	email	
string	primary_address	
string	profile_image	

Service_request		
integer	service_request_id	PK
integer	service_id	FK
string	customer_id	FK
string	professional_id	FK
datetime	date_of_request	
datetime	date_of_completion	
enum	service_status	
string	remarks	
integer	ratings	

Fig - ER-Diagram

Presentation Link - https://drive.google.com/file/d/1OZIVqsaTFk0nOt1b2PPfeR_MGIhhBBea/view?usp=sharing