



HomeMate

PROJECT REPORT

Presented By:

Aayansh Yadav




Roll Number:

23F2003711





About Me

New Tab	
Aayansh Yadav	
Name	Aayansh Yadav
Roll Number	23F2003711
Student Mail	23F2003711@ds.study.iitm.ac.in
  	@Noctivagous

Project Details

HomeMate: Household Services Application



The Household Services Application acts as a platform connecting customers with service professionals for home services. Customers can create service requests for tasks like plumbing, AC servicing, or electrical work, and professionals can accept these requests based on their expertise. The admin oversees the entire system, managing services, approving professionals, and ensuring smooth operations. Service professionals can track their jobs, receive reminders, and update their service statuses. This creates an efficient, organized system where customers can easily book and manage services, and professionals can find consistent work while maintaining high standards through customer feedback.



TECHNOLOGIES USED

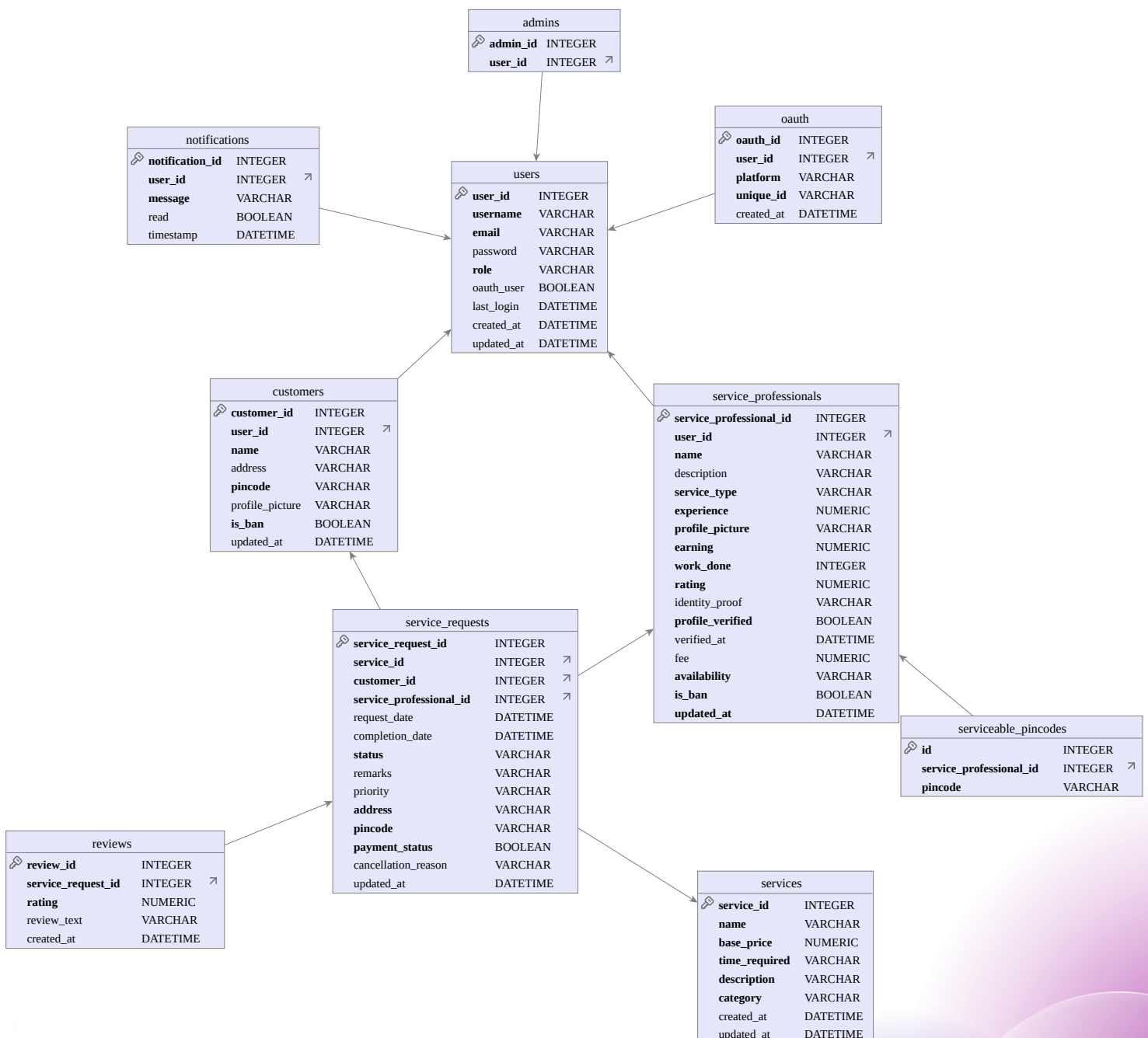
- **Flask:** Lightweight Python web framework to handle the backend API.
- **VueJS:** JavaScript framework for building the user interface.
- **Bootstrap:** CSS framework for responsive design and styling.
- **Flask-SQLAlchemy:** ORM for managing database interactions.
- **Flask-RESTful:** For creating RESTful APIs.
- **Flask-JWT-Extended:** Handles authentication with JWT tokens.
- **Flask-Caching:** Adds caching for improved performance.
- **Celery:** Asynchronous task queue for background jobs like reminders and monthly reports.
- **Redis:** Caching layer and supports background job management.
- **SQLite:** Lightweight database for storing user data and service requests.
- **WeasyPrint:** Used for generating PDF reports.
- **Razorpay:** Payment gateway for handling transactions.



DB Schema



The database schema for Household Services Application includes tables for users, admins, customers, service professionals, services, requests, reviews, and notifications. It connects these entities to manage roles, service requests, and reviews efficiently, enabling smooth workflow and data management.





Directory Structure



```
├── HomeMate
│   ├── client
│   │   ├── index.html
│   │   ├── jsconfig.json
│   │   ├── package.json
│   │   ├── package-lock.json
│   │   ├── README.md
│   │   ├── src
│   │   │   ├── App.vue
│   │   │   ├── assets
│   │   │   ├── components
│   │   │   ├── main.js
│   │   │   ├── router
│   │   │   ├── stores
│   │   │   ├── utils.js
│   │   │   └── views
│   │   └── vite.config.js
│   ├── server
│   │   ├── app
│   │   │   ├── api
│   │   │   ├── config.py
│   │   │   ├── custom.py
│   │   │   ├── __init__.py
│   │   │   ├── models
│   │   │   ├── static
│   │   │   ├── tasks
│   │   │   └── templates
│   │   ├── instance
│   │   │   └── HomeMate.db
│   │   ├── requirements.txt
│   │   └── run.py
│   └── Project Report.pdf
```



API Resource



Authentication & Authorization

- Signup: /api/signup
- Login: /api/login
- Google OAuth Signup: /api/signup/google-oauth
- Google OAuth Login: /api/login/google-oauth
- Facebook OAuth Signup: /api/signup/facebook-oauth
- Facebook OAuth Login: /api/login/facebook-oauth
- Token Check: /api/token-checker
- Logout: /api/logout
- Reset Password: /api/reset-password

User Profile & Management

- Profile: /api/profile
- Customer: /api/customers
- Servicer: /api/servicer

Services & Requests

- Service: /api/services
- Service Request: /api/service-requests
- Review: /api/review

Utility & Data

- AutoComplete: /api/autocomplete
- Pincode to City: /api/pincode-to-city
- Stats: /api/stats

Payments & Transactions

- Payment: /api/payment

Tasks

- Tasks: /api/tasks



Presentation Video



Youtube : <https://youtu.be/uhcoqsnWUHc>

Google Drive :

https://drive.google.com/file/d/1ndlJo6A4w5kCxLul3AgOrtrkQtFZsxQn/view?usp=drive_link



**Thank You
For Your Attention!**

Aayansh Yadav

29 Nov 2024

