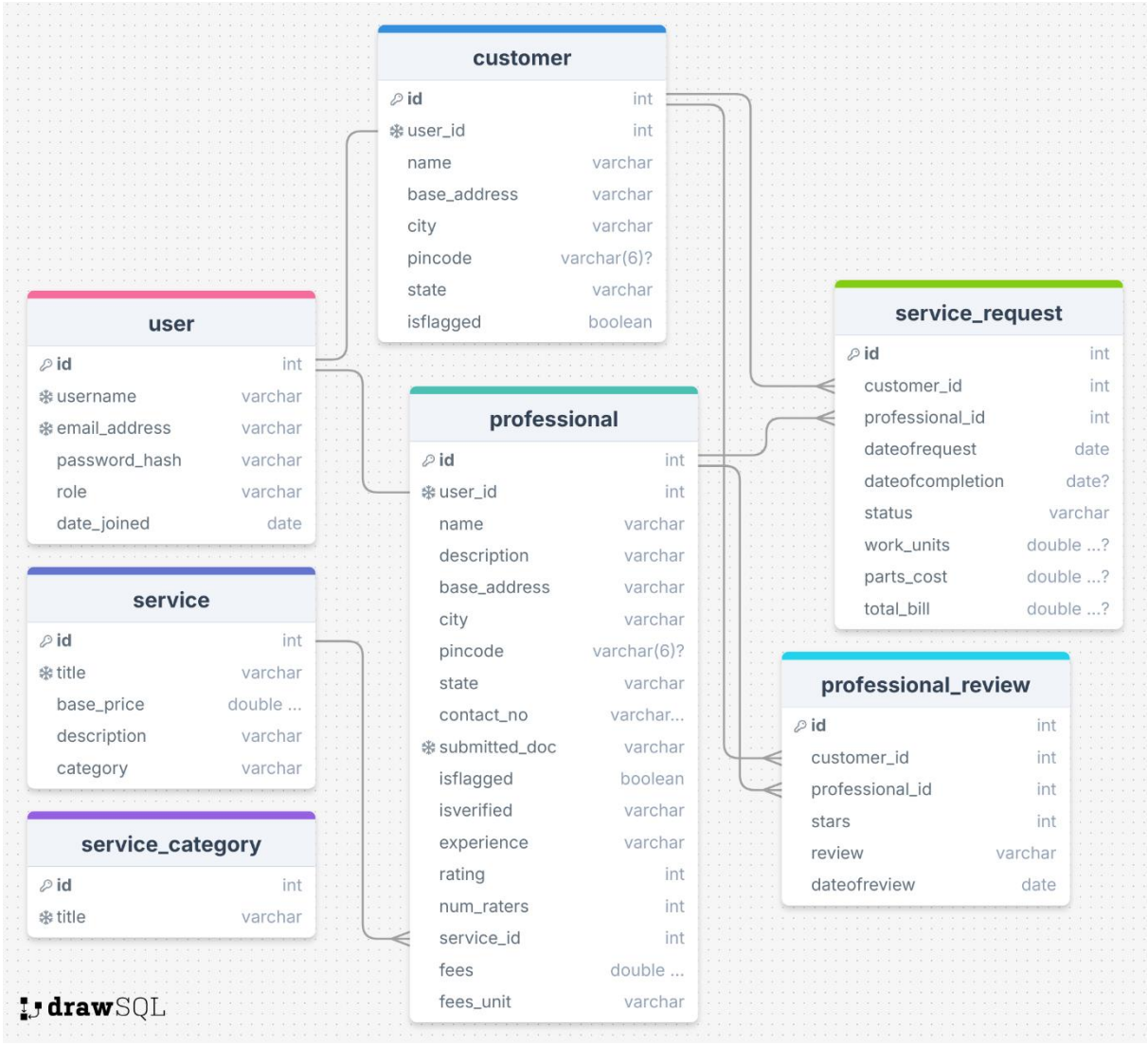
Project Report

- **About the Author:**
  Name: Bhavya Goyal, Roll No: 23F1002830, Email: 23f1002830@ds.study.iitm.ac.in

- **Project Description:**
  This project is made for Modern Application Development –2 Project Course, (Subject code: BSCS2006P) in September, 2024 term.
  The project is a full-stack web platform, which aims to connect Customers and Service Professionals together, for easy home services booking.

- **Technology Used:**
  1. Python Flask Framework –Backend framework on which the application is built on.
  2. SQLite – Database used for the application.
  3. Flask-Sqlalchemy – ORM for the database, in the flask application.
  4. Flask RestX- for Api resources making.
  5. Flask JWT extended & Flask-bcrypt – For secure login functionality and password hashing and RBAC.
  6. Flask SSE, Flask caching - For Server sent events and caching.
  7. Celery – Backend asynchronous jobs.
  8. Redis and MailHog – Message brokers, Result queue for backend jobs and local SMTP server.
  9. Gunicorn, gevent – Running several instances of backend application.
  10. Vue JS – Frontend of the application (along with HTML, CSS).
  11. Vue Router, Pinia – Client-side routing and State management.

- **Project Architecture:**
  1. Bash files to setup/run the application along with git configuration files and original python files to run the app, in the main folder.
  2. Backend folder contains python code for backend, requirements.txt, instance folder for the database. The ‘homemate’ directory is a python package containing code for the application. It has several sub-directories for various uses, like ‘models’ containing the ORM tables, ‘auth’ containing api resources made for authentication and so on...
  3. Frontend folder contains Vue-cli generated project template for a SPA. The ‘src’ folder contains most of the code. It has ‘views’ which represent entire pages on client side and ‘components’ which are independent, individual units on a page. There are few ‘composables’ that are JS functions common to all components. Further, there is a vue-router file and ‘stores’ directory containing pinia-stores for the frontend application. The is a global assets folder that contains logo, favicon and the common CSS to all pages.
- **Database Schema:**



- **Project Features:**
    1. Customer – Can login, search for services and professionals, make service requests, cancel them, make payment when served, review professionals, and view statistics. He receives a monthly report on mail too, of all requests he has booked in that month.
    2. Service Professional – Can login but requires document verification from admin to use the platform, accept/reject service requests, Serve the request (by completing it), monitor statistics. He receives a mail daily holding information of pending and due-tomorrow service requests.
    3. Admin – View Platform statistics, Approve or reject the documents of Service professional, flag/unflag users and professionals, add/edit/delete services available on the platform, request a CSV copy of all service requests on the platform.

- **Video Link:**

 https://drive.google.com/file/d/1HiIV1vl9spexhCnBAD6twbvjJUy08QDm/view?usp=sharing