# MAD-1 PROJECT REPORT

## HOUSEHOLD SERVICES APPLICATION: BY: ACHARYA SAINYAM

23f1002391@ds.study.iitm.ac.in

PROBLEM STATEMENT:

It is a multi-user app (requires one admin and other service professionals/ customers) which acts as platform for providing comprehensive home servicing and solutions.

TECHNOLOGIES AND FRAMEWORKS USED:

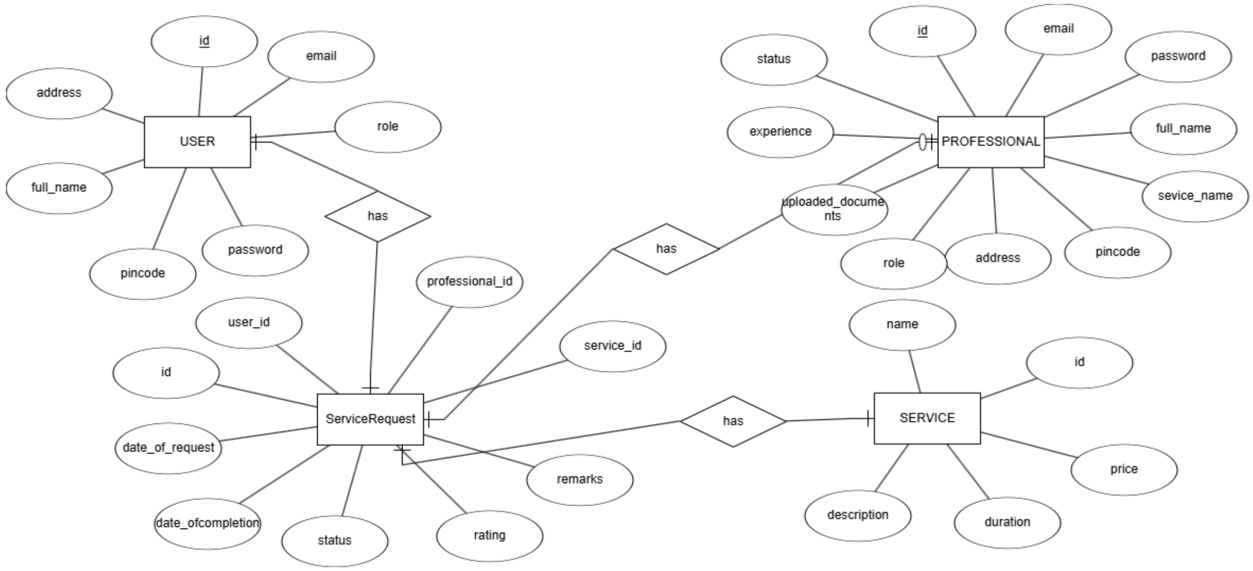
- Python: programming language to drive the development stack
- HTML: for developing the web page
- Bootstrap: to make the frontend appealing and easily navigate
- Jinja: used for separation of logic and dynamic HTML rendering in the Flask app
- SQLite: database for storing users, professionals, services, service_requests relational data
- Flask: a web framework for building the web application backend
- Flask-SQLAlchemy

## ER DIAGRAM:

1 – Users(id INTEGER PRIMARY_KEY, email STRING, password STRING, full_name STRING, address STRING, pincode STRING, role INTEGER default=1)
2 – Professional(id INTEGER PRIMARY_KEY, email STRING, password STRING, full_name STRING, address STRING, pincode STRING, service_name STRING , uploaded_documents STRING, experience INTEGER, status STRING, role INTEGER default=2)
3 – Service(id INTEGER PRIMARY_KEY, name STRING, price FLOAT, duration INTEGER, description STRING)
4 -- ServiceRequest(id INTEGER PRIMARY_KEY, user_id INTEGER FOREIGN_KEY, professional_id INTEGER FOREIGN_KEY, service_id INTEGER FOREIGN_KEY, date_of_request DATETIME, date_of_completion DATETIME, service_status STRING, rating INTEGER, remarks STRING)

### Relationships:

1. **User → ServiceRequest**
2. **Professional → ServiceRequest**
3. **Service → ServiceRequest**
4. **Professional → Service**



## API ENDPOINTS:

### Service API Endpoints

- **Get all services**: GET /api/get_services
- **Add a new service**: POST /api/add_services
- **Edit a service**: PUT /api/edit_services/<id>
- **Delete a service**: DELETE /api/delete_services/<id>

### Professional API Endpoints

- **Get all professionals**: GET /api/get_professionals
- **Add a new professional**: POST /api/add_professionals
- **Edit a professional**: PUT /api/edit_professionals/<id>
- **Delete a professional**: DELETE /api/delete_professionals/<id>

### User API Endpoints

- **Get all users**: GET /api/get_users
- **Add a new user**: POST /api/add_users
- **Edit a user**: PUT /api/edit_users/<id>
- **Delete a user**: DELETE /api/delete_users/<id>

**VIDEO** https://drive.google.com/drive/folders/1jxeGNoKOOzg9qA48S4Djks5K8q5F0DXW