Project Report for Household service App

(Modern Application Development - I)

Author
Md Anas Alam
22F3000639
22f3000639@ds.study.iitm.ac.in

I am a data science enthusiast. I enjoy coding and making softwares. This project was extremely enjoyable to create although I tackled many obstructions while creating this project but eventually it helped me in gaining more in depth knowledge about the technologies used in this project.

Description:

This project is a multi user and multi roles web application. I have used HTML, CSS, Flask, SQLAlchemy, jinja template and Matplotlib. In this project I have created three roles i.e admin, customer and service professional. This is a web application where the admin has the control to create a service for all the users. Customers can opt for a service created by the admin and send service requests to the service professional. This is the detailed functionality of each of the given roles i.e admin, customer and service professional.

Admin: Admin can create and edit a service request with the name of the service, description of the service and base price. Admin also has the ability to delete a particular service. Admin can see all the logged in user and service professionals and has the ability to block a user and service professional. Admin can also visualize the total revenue generated by each type of service being offered.

Customer: Customer can see all the available services created by the Admin and opt for a particular service. Customer has the ability to edit and delete a service request made by the customer. Customers can close the service request upon completion of the requested service and provide rating and feedback.

Service professional: Service professionals can see all the available service requests made by the customer and accept the service request by their convenience.

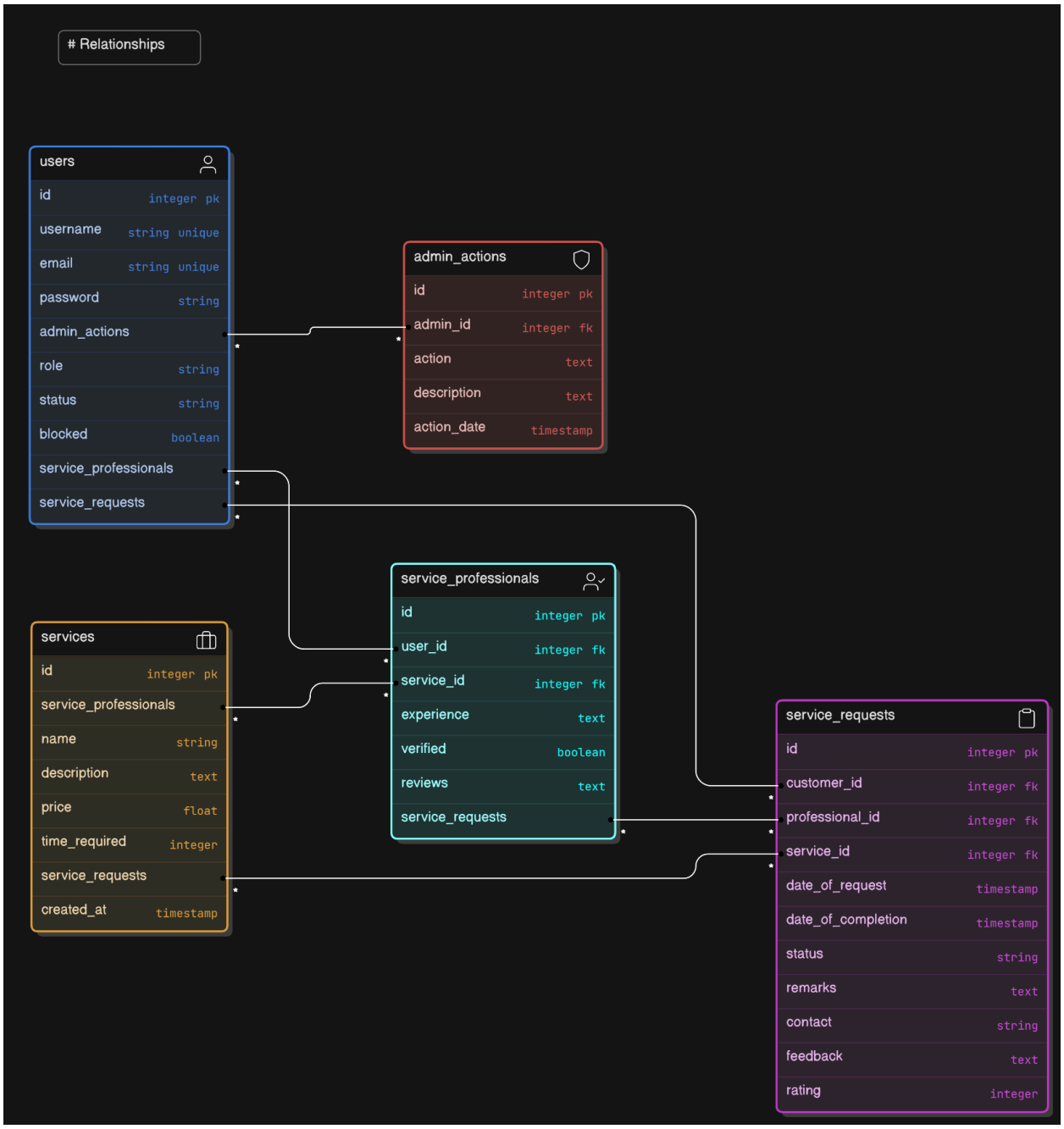
Technologies Used:

Flask: A Python web framework used to build the web application helping in development of the web application and web pages.
Flask-SQLAlchemy: An extension for Flask that integrates SQLAlchemy, simplifying database handling and seamless interaction with the app's database.
Matplotlib:A python framework for creating interactive visualization.
Datetime:module to store data and time of completing service request..
Integrity error: Used for handling errors.
Flask session: A Flask feature used to manage and store data.
Flask flash: Used for one time flashing messages for validation.
`redirect` and `url_for`: Used for navigation purposes and validating web pages.
Render Templates: Used for connecting backend logic with frontend designs.

DB Schema Design:

Relationship:

- Service has one-to-many with Service request i.e each service can have multiple requests. Service has One-to-Many with Service Professional i.e each service can have multiple professionals. Service has One-to-Many with service request (each service can have multiple requests).
- Service professional has Many-to-One with User and service i.e (each professional is linked to one user and one service) and Service professional has One-to-Many with service request i.e (each professional can handle multiple requests).
- Users have a one-to-many relationship with service requests (one user can make multiple requests) and Users have a one-to-many relationship with service professionals (each user can have one or more service professional roles).
- Service request has Many-to-One relationship with User, Service professional and services i.e (each request is made by one customer, each request is handled by one professional and each request corresponds to one service)



Routes samples:

```
@app.route('/signup', methods=['GET', 'POST'])
@app.route('/login', methods=['GET', 'POST'])
@app.route('/admin_dashboard/<int:curr_login_id>', methods=['GET'])
@app.route('/customer_dashboard/<int:curr_login_id>', methods=['GET'])
@app.route('/professional_dashboard/<int:curr_login_id>', methods=['GET'])
```

Video Link:
Drive link