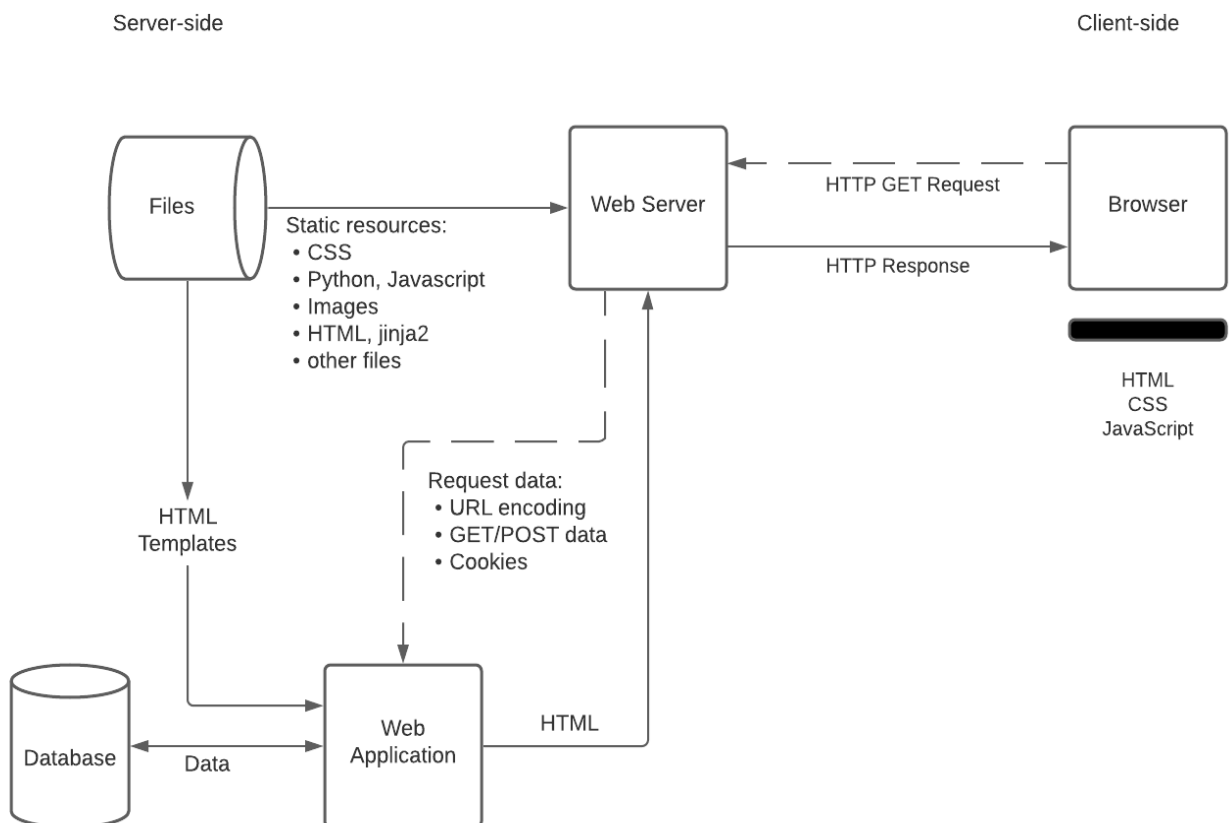


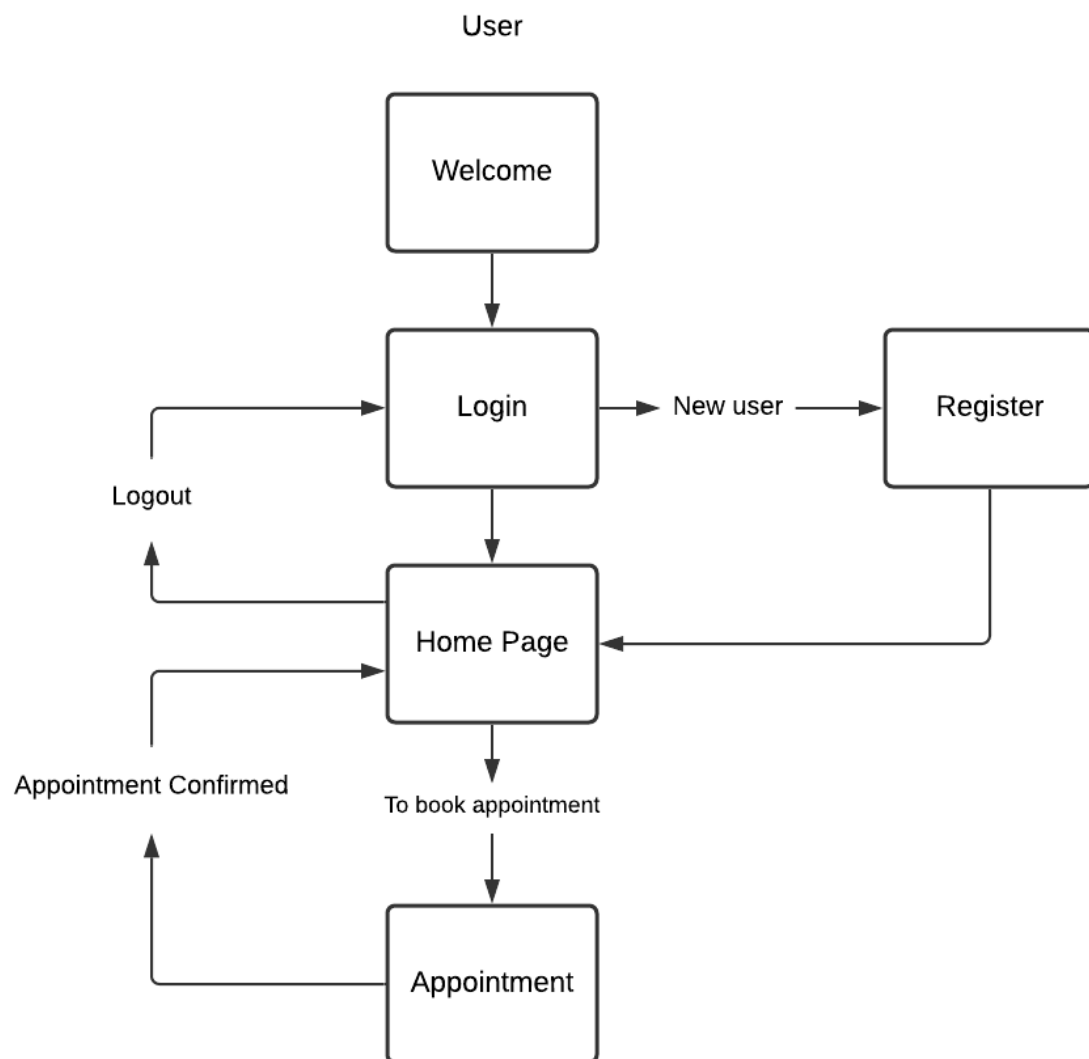
# Architecture

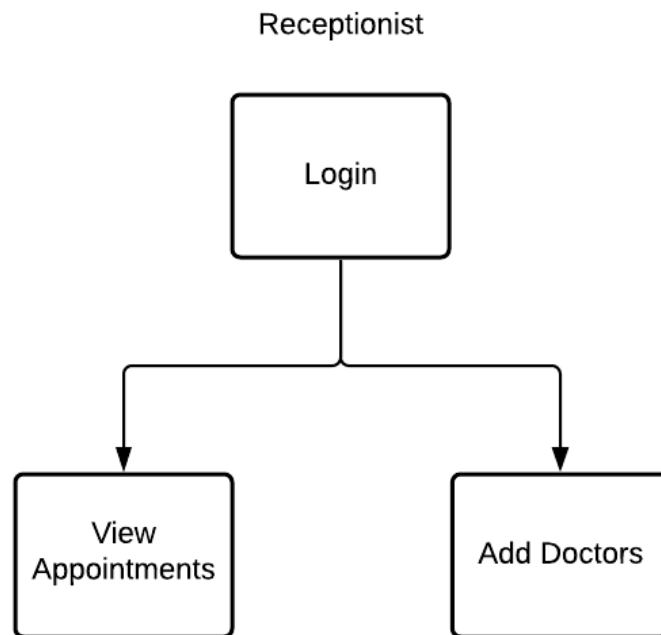
## Overview

- We are deploying a website to help users make Hospital appointment booking hassle free.
- Users can filter the doctors by specialization and experience.
- Users can choose a date and time of the appointment at their convenience.
- The User can also book the appointment for others who don't have an account on our website.



- Simple navigation chart of our website





## Frontend

- Welcome
  - A brief introduction about the Clinic
  - Option to Sign in
- Login
  - Users can sign in using Phone Number and Password.
  - New users can choose the Sign Up option.
- Sign Up
  - Users can create an account using a Phone number along with their Name, Date of Birth and Gender.
- Home Page
  - Users can see the information regarding upcoming appointments
  - Book an Appointment

- Information regarding the services offered
  - Edit Profile, Change Password and Logout
- Appointment
  - List of all doctors available with details with filters
  - Availability of doctors
  - Appointment for self or for others.
- Login(as Receptionist)
  - Receptionists can login using Phone Number and Password.
- Login(as Admin)
  - Administrators can login using Phone Number and Password.
- Receptionist Page
  - Receptionist can see the information regarding all upcoming appointments
  - Walk-in Appointments
- Admin
  - Admin can add Doctors, add Receptionist, add Admin or edit their details.
- Doctor
  - Doctors can prescribe medicines to the patient.

## **Backend**

- Home Page
  - /home

- We will give the Patient ID to this as an incoming parameter
- Login
  - /login
  - This will send Login credentials to the backend to authenticate.
- Sign Up
  - /signup
  - This will send all the user data to the backend to create a record in the database using POST request.
- Appointment Page
  - /aptmnt
  - This will render the html using jinja2 depending upon the user input like sort or group the Doctors
  - This takes the Filters information as the input

## Databases

### Tables

- Patients
  - To verify the login credentials and to autofill the details if the appointment is for self

- Schema-Patients(phno char(10),password varchar(20),first name varchar(20), last name varchar(20) ,dob date, gender char(1))
- Appointments
  - Will have the details of which patient is consulting which doctor at what time and date
  - Schema-aptmnt(apmnt\_id int, patient\_id varchar(40),doctor\_id varchar(20),aptmnt\_time time,aptmnt\_date date)
- Doctors
  - Doctors profile and their availability
  - Schema-Doctors(Doctor\_id varchar(20), doctor\_first\_name varchar(20),doctor\_last\_name varchar(20),doctor\_specialization varchar(20),doctor\_experience int,doctor\_education,doctor\_gender char(1) ,doctor\_image varchar(20))
- Doctor\_availability
  - When will the doctor be available

- Schema- Doctor\_availability(doctor\_id varchar(20),monday bool, tuesday bool,wednesday bool,thursday bool,friday bool,saturday bool,sunday bool)
- Receptionists
  - Credentials of Receptionists
  - Schema- Receptionist(recp\_id varchar(20),recp\_id password varchar(20),recp\_name varchar(20),recp\_phno int(10))
- Admin
  - Credentials of Admin
  - Schema-Admin(Admin\_id varchar(20), admin\_password varchar(20),admin\_first\_name varchar(20) admin\_last\_name(20))
- Prescription
  - Prescription of Patients
  - Schema- Prescription(Prescription\_id varchar(20),prescription\_notes varchar (200),patient\_id varchar(20),doctor\_id varchar(20),pres\_date date)