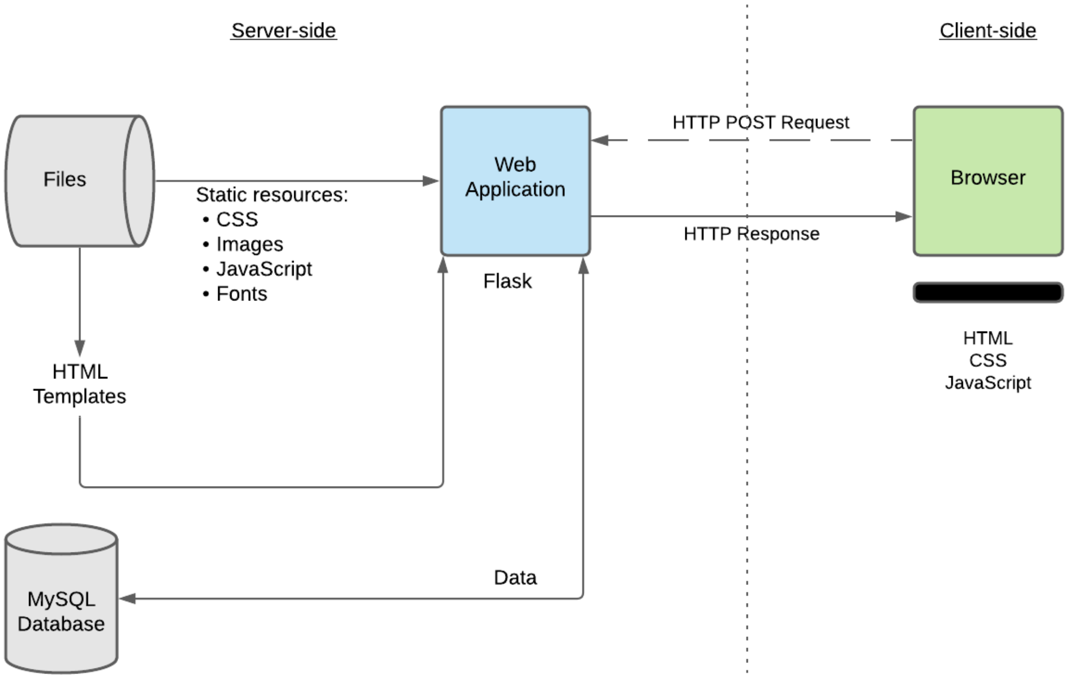
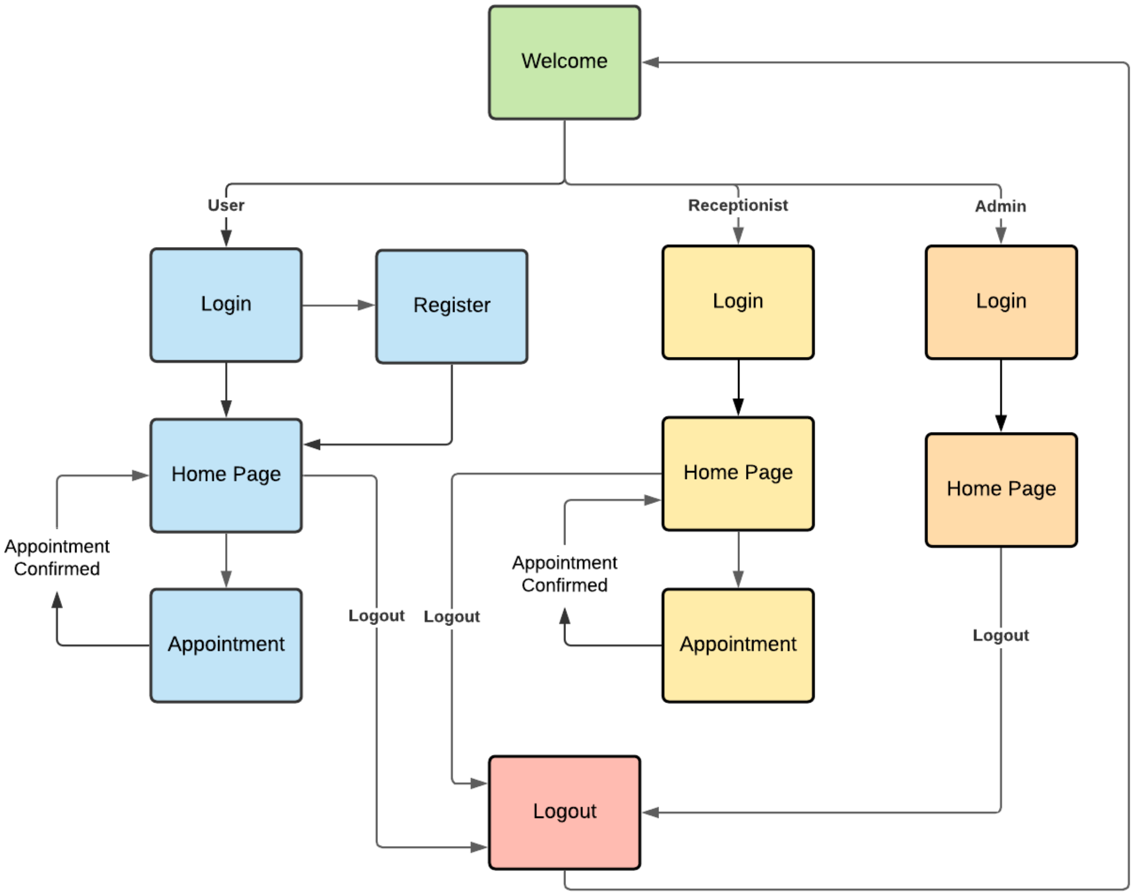
Architecture

Overview

- Nova Clinic is a web application that enables Health Centers to provide users an easy way of booking a doctor's appointment online and helps the Health Centers to manage the appointments with ease.
- Architecture



- Navigation chart of website



Frontend

User Portal

- Welcome
    - A brief introduction about the Clinic
    - Redirects to User Portal or Receptionist or Admin portal

- Login
    - Users can sign in using their Phone Number and Password.
    - New users can choose the Sign Up option.

- Sign Up
    - Users can create an account using their Phone number along with their Name, Date of Birth and Gender.
- Home Page
    - Users can view their upcoming appointments
    - Users can choose to book an Appointment
    - Users can choose to cancel an Appointment
    - Users can edit their profile, change their password and can logout
    - Information regarding the services offered
- Appointment
    - List of all Doctors information with filters
    - Availability of doctors slots on the selected date is shown

Receptionist Portal

- Login
    - Receptionists can log in using their Username and Password
- Receptionist Home Page
    - Receptionists can view all upcoming appointments
    - Receptionists can book a Walk-In Appointment
- Appointment (Walk-In Appointments)
    - Receptionists can book an appointment for patients with no account.
    - List of all Doctors information with filters
    - Availability of doctors slots on the selected date is shown

Admin Portal

- Login
    - Admins can log in using their Username and Password.
- Admin Home Page
    - Admin can add, update and delete Doctors
    - Admin can add and delete Receptionists
    - Admin can view all the existing Doctors, Receptionists and Admins

Backend

- /login
    - Receives the Phone Number and Password through a POST request
    - If the credentials are correct, This will redirect to the Homepage, else displays the error flash message
- /signup
    - Receives the User Information like Phone Number, Name, Date of Birth, Gender and Password
    - Verifies whether the given Phone Number is new and few other inputs compatibility checks.
    - If everything entered was correct, This'll add the User to the Database.
- /userName
    - Receives the Phone Number of the logged-in user
    - Returns the Name of the User

- /home
    - Receives the Phone Number of the logged-in user
    - Renders the User’s homepage along with the data of upcoming appointments and the name of the User.
- /updateInfo
    - Receives the Edited Phone Number, First Name and Last Name
    - Updates the User Information in the database after a few verifications
- /updateCredentials
    - Receives the Old Password and New Password
    - Updates the User Credentials in the database if the old password matches the password in the database
- /aptmnt
    - Receives the Doctor filter choices
    - Renders the Appointment page accordingly
- /getslotsinfo
    - Receives the Date and Doctor ID
    - Returns the data of the Slots of that Doctor on that date.
- /confirmaptmnt
    - Receives the Date, Doctor ID, Name of the Patient, Slot
    - Creates the appointment in the Database and updates the Slots data in the Database
- /aptmntDelete
    - Receives the appointment ID, date, slot, Doctor ID
    - Deletes the appointment in the Database and updates the Slots data in the Database

- /rlogin
    - Receives the Receptionists Username and Password
    - If the credentials are correct, This will redirect to the Receptionists Homepage, else displays the error flash message
- /receptionist
    - Receives the Filter choices of Appointments
    - Renders the Receptionist Home Page accordingly
- /raptmnt
    - Receives the Filter choices of Doctor
    - Renders the Appointment page accordingly
- /alogin
    - Receives the Admins Username and Password
    - If the credentials are correct, This will redirect to the Admins Homepage else displays the error flash message
- /admin
    - Returns the IDs and Name of the existing Doctors, Receptionists and Admins
- /addDoctor
    - Receives the Doctor ID, Name, Specialization, Gender, Experience and Education of the Doctor to be added
    - Will verify if the Doctor ID is unique, If unique, will add the Doctor to the Database.

- /updateDoctor
  - Receives the Doctor ID, Name, Specialization, Gender, Experience, Education of the Doctor to be updated
  - Will update the Doctor information in the Database
- /deleteDoctor
  - Receives the Doctor ID of the Doctor to be deleted
  - Will delete the Doctor information in the Database
- /addReceptionist
  - Receives the Receptionist ID, Name of the Receptionist to be added
  - Will verify if the Receptionist ID is unique, If unique will add the Receptionist to the Database.
- /deleteReceptionist
  - Receives the Receptionist ID of the Receptionist to be deleted
  - Will delete the Receptionist information in the Database
- /logout
  - Will clear the session variable and will redirect to the Welcome Page

Database

Tables and their Schema

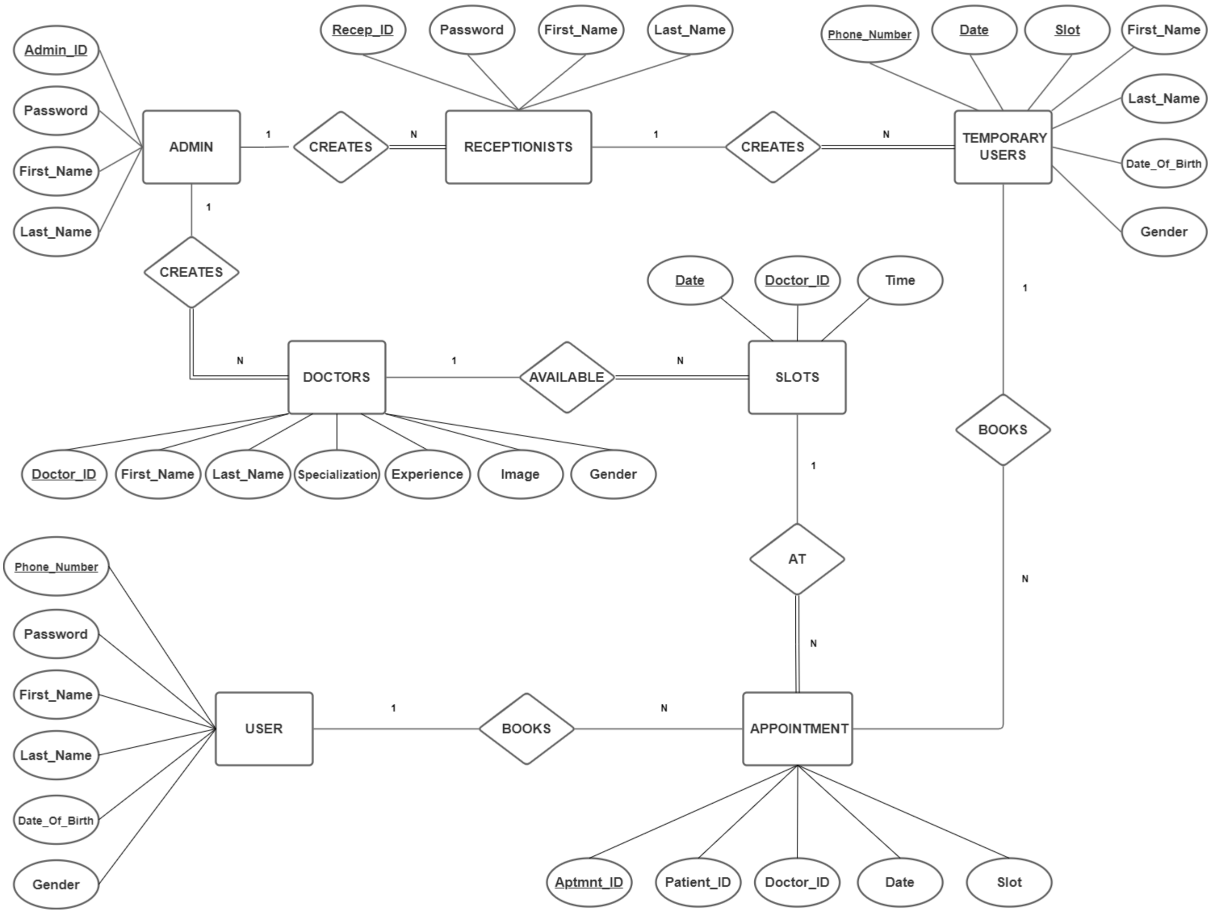
- Users
  - To store the information and credentials of the user
  - users(`phno` char(10) NOT NULL, `password` char(56) NOT NULL, `FName` varchar(20) NOT NULL, `LName` varchar(20) NOT NULL, `dob` date NOT NULL, `gender` varchar(6) NOT NULL, PRIMARY KEY (`phno`));
- Appointments
  - To store the details of which patient is consulting which doctor at what time(slot) and date
  - aptmnt(`aptmnt_id` int(11) NOT NULL AUTO_INCREMENT, `patient_id` char(10) NOT NULL, `doctor_id` varchar(10) NOT NULL, `date` date NOT NULL, `slot` char(11) NOT NULL, PRIMARY KEY (`aptmnt_id`), CONSTRAINT `aptmnt_ibfk_2` FOREIGN KEY (`doctor_id`) REFERENCES `doctors` (`doctor_id`));
- Doctors
  - To store the information of Doctors
  - Doctors(`doctor_id` varchar(10) NOT NULL, `FName` varchar(20) NOT NULL, `LName` varchar(20) NOT NULL, `doctor_specialization` varchar(20) NOT NULL, `doctor_experience` int(11) NOT NULL, `gender` varchar(6) NOT NULL, `doctor_education` varchar(40) NOT NULL,

`doctor_image` varchar(20) NOT NULL, PRIMARY KEY (`doctor_id`));

- Slots
    - To store the availability status of the doctor at a given date, time
    - slots(`date` char(10) NOT NULL,`doctor_id` varchar(10) NOT NULL,`time` char(24) NOT NULL DEFAULT '000000000000000000000000',PRIMARY KEY (`date`,`doctor_id`),KEY `doctor_id` (`doctor_id`),CONSTRAINT `slots_ibfk_1` FOREIGN KEY (`doctor_id`) REFERENCES `doctors` (`doctor_id`));
- Receptionists
    - To store the Name and Credentials of Receptionists
    - receptionists(`recep_id` char(10) NOT NULL, `FName` varchar(20) NOT NULL, `LName` varchar(20) NOT NULL, `password` char(56) NOT NULL DEFAULT '43c21023f40197a9e0e122d3d191fb2c101f664bf4a1cb4ca886dff7', PRIMARY KEY (`recep_id`));
- Admin
    - To store the Name and Credentials of Admins
    - admin(`admin_id` varchar(10) NOT NULL, `password` char(56) NOT NULL, `FName` varchar(20) NOT NULL, `LName` varchar(20) NOT NULL, PRIMARY KEY (`admin_id`));

- Temporary Users
  - To store the details of Patients who booked appointments through walk-in appointments
  - temp_users(`FName` varchar(20) NOT NULL, `LName` varchar(20) NOT NULL, `dob` date NOT NULL, `gender` varchar(6) NOT NULL, `phno` char(10) NOT NULL, `slot` char(11) NOT NULL, `date` date NOT NULL, PRIMARY KEY (`slot`,`date`,`phno`));

ER Diagram



Admin_ID
Password
First_Name
Last_Name
ADMIN
1
CREATES
N
Recep_ID
Password
First_Name
Last_Name
RECEPTIONISTS
1
CREATES
N
Phone_Number
Date
Slot
First_Name
Last_Name
Date_Of_Birth
Gender
TEMPORARY USERS
1
CREATES
N
DOCTORS
Doctor_ID
First_Name
Last_Name
Specialization
Experience
Image
Gender
1
AVAILABLE
N
Date
Doctor_ID
Time
SLOTS
1
AT
N
1
BOOKS
N
Phone_Number
Password
First_Name
Last_Name
Date_Of_Birth
Gender
USER
1
BOOKS
N
APPOINTMENT
Aptmnt_ID
Patient_ID
Doctor_ID
Date
Slot