# Assignment 2(Heal At Home)

**Team Members: (EAD128)**
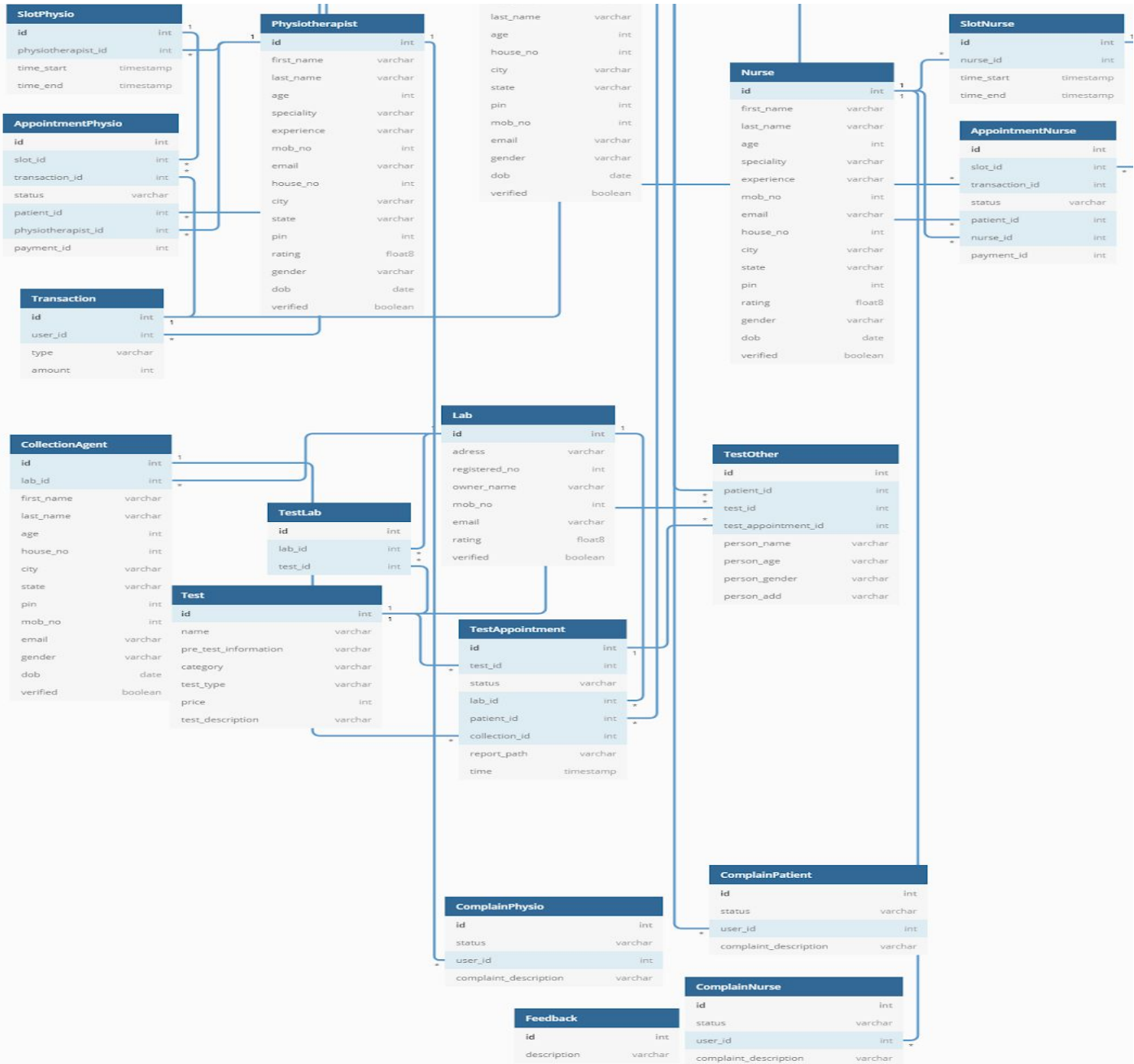Aman Kumar (S20170010010)
Shivam Gupta (S20170010149)
Kalpajyoti Handique (S20170010061)
Siddhartha Goswami (S20170010152)

# Database Schema

**SlotPhysio**

| id | int |
|---|---|
| physiotherapist_id | int |
| time_start | timestamp |
| time_end | timestamp |

**AppointmentPhysio**

| id | int |
|---|---|
| slot_id | int |
| transaction_id | int |
| status | varchar |
| patient_id | int |
| physiotherapist_id | int |
| payment_id | int |

**Transaction**

| id | int |
|---|---|
| user_id | int |
| type | varchar |
| amount | int |

**Physiotherapist**

| id | int |
|---|---|
| first_name | varchar |
| last_name | varchar |
| age | int |
| speciality | varchar |
| experience | varchar |
| mob_no | int |
| email | varchar |
| house_no | int |
| city | varchar |
| state | varchar |
| pin | int |
| rating | float8 |
| gender | varchar |
| dob | date |
| verified | boolean |

| last_name | varchar |
|---|---|
| age | int |
| house_no | int |
| city | varchar |
| state | varchar |
| pin | int |
| mob_no | int |
| email | varchar |
| gender | varchar |
| dob | date |
| verified | boolean |

**Nurse**

| id | int |
|---|---|
| first_name | varchar |
| last_name | varchar |
| age | int |
| speciality | varchar |
| experience | varchar |
| mob_no | int |
| email | varchar |
| house_no | int |
| city | varchar |
| state | varchar |
| pin | int |
| rating | float8 |
| gender | varchar |
| dob | date |
| verified | boolean |

**SlotNurse**

| id | int |
|---|---|
| nurse_id | int |
| time_start | timestamp |
| time_end | timestamp |

**AppointmentNurse**

| id | int |
|---|---|
| slot_id | int |
| transaction_id | int |
| status | varchar |
| patient_id | int |
| nurse_id | int |
| payment_id | int |

**CollectionAgent**

| id | int |
|---|---|
| lab_id | int |
| first_name | varchar |
| last_name | varchar |
| age | int |
| house_no | int |
| city | varchar |
| state | varchar |
| pin | int |
| mob_no | int |
| email | varchar |
| gender | varchar |
| dob | date |
| verified | boolean |

**TestLab**

| id | int |
|---|---|
| lab_id | int |
| test_id | int |

**Test**

| id | int |
|---|---|
| name | varchar |
| pre_test_information | varchar |
| category | varchar |
| test_type | varchar |
| price | int |
| test_description | varchar |

**Lab**

| id | int |
|---|---|
| adress | varchar |
| registered_no | int |
| owner_name | varchar |
| mob_no | int |
| email | varchar |
| rating | float8 |
| verified | boolean |

**TestAppointment**

| id | int |
|---|---|
| test_id | int |
| status | varchar |
| lab_id | int |
| patient_id | int |
| collection_id | int |
| report_path | varchar |
| time | timestamp |

**TestOther**

| id | int |
|---|---|
| patient_id | int |
| test_id | int |
| test_appointment_id | int |
| person_name | varchar |
| person_age | varchar |
| person_gender | varchar |
| person_add | varchar |

**ComplainPhysio**

| id | int |
|---|---|
| status | varchar |
| user_id | int |
| complaint_description | varchar |

**ComplainPatient**

| id | int |
|---|---|
| status | varchar |
| user_id | int |
| complaint_description | varchar |

**ComplainNurse**

| id | int |
|---|---|
| status | varchar |
| user_id | int |
| complaint_description | varchar |

**Feedback**

| id | int |
|---|---|
| description | varchar |

dbdiagram.io

## Reason behind using the database

- Data used in our project is mostly relational therefore we have used RDBMS
- It can handle a lot more data, its highly scalable
- PostgreSQL is fully ACID Compliant
- It has multi-version concurrency control (MVCC) features

## DBMS Scalability and usability analysis

**PostgreSQL can handle a lot of data. The current posted size limits are listed below:**

| Limit | Value |
|---|---|
| Maximum Database Size | Unlimited |
| Maximum Table Size | 32 TB |
| Maximum Row Size | 1.6 TB |
| Maximum Field Size | 1 GB |
| Maximum Rows per Table | Unlimited |
| Maximum Columns per Table | 250 - 1600 depending on column types |
| Maximum Indexes per Table | Unlimited |

**Module implementation with test cases**

The module implemented for this section is the physiotherapist portal. This module mainly includes:
- Profile management for physiotherapist
- Accepting booking requests
- Making slots available for the patient
- Managing appointments

In the implementation of the module, we can see the profile of the physiotherapist.
This is the homepage of physiotherapist

Then give the ability to modify the profile of the physiotherapist himself.
(updating his experience to 15 years)



Changes are made to his actual profile being shown at his home page

Then the physiotherapist slot creation is enabled, so he can himself decide when to attend the patients



After that the physiotherapist can see the list of his slots.

| Date | Time | Slot Status |
|------|------|-------------|
| April 11, 2020 | 09:00:00 | Not Booked |
| April 10, 2020 | 18:00:00 | Not Booked |
| April 10, 2020 | 16:00:00 | Booked |
| April 10, 2020 | 13:00:00 | Not Booked |
| April 9, 2020 | 15:00:00 | Not Booked |
| April 9, 2020 | 17:00:00 | Booked |
| April 9, 2020 | 16:00:00 | Not Booked |
| April 9, 2020 | 13:00:00 | Not Booked |
| April 9, 2020 | 12:00:00 | Booked |

After the slots are made and the physiotherapist can view his bookings, and accordingly, he can see the list of bookings

**HEAL AT HOME**

## Dr. Aman Kumar

| Patient Name | Patient Address | Date | Time | Attended |
|---|---|---|---|---|
| Kalpo | Dibrugarh, Assam | April 10, 2020 | 16:00:00 | Upcoming |
| Sanjay | Deoria, Uttar Pradesh | April 10, 2020 | 18:00:00 | Upcoming |
| Sanjay | Deoria, Uttar Pradesh | April 11, 2020 | 09:00:00 | Upcoming |
| Shivam | Haridwar, Uttarakhand | April 9, 2020 | 13:00:00 | Completed |