

--User table for general user information

```
CREATE TABLE User (  
    user_id INT PRIMARY KEY AUTO_INCREMENT,  
    username VARCHAR(50) NOT NULL,  
    password VARCHAR(255) NOT NULL,  
    email VARCHAR(100) UNIQUE NOT NULL,  
    profile_picture VARCHAR(255), -- URL or path to the profile picture  
    role ENUM('Admin', 'Caregiver', 'Consultant', 'Careseeker') NOT NULL,  
    profile_info TEXT,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE  
    CURRENT_TIMESTAMP  
);
```

--Caregiver table with a foreign key reference to User

```
CREATE TABLE Caregiver (  
    caregiver_id INT PRIMARY KEY,  
    FOREIGN KEY (caregiver_id) REFERENCES User(user_id) ON DELETE CASCADE,  
    experience TEXT,  
    availability_status BOOLEAN,  
    region VARCHAR(100), --store caregiver's region for location-based display  
    rating DECIMAL(3, 2) DEFAULT 0, -- Average rating for display  
    bio TEXT  
);
```

--Careseeker table with a foreign key reference to User

```
CREATE TABLE Careseeker (  
    careseeker_id INT PRIMARY KEY,  
    FOREIGN KEY (careseeker_id) REFERENCES User(user_id) ON DELETE CASCADE,  
    medical_info TEXT,  
    address VARCHAR(255),  
    contact_info VARCHAR(50),  
    rating DECIMAL(3, 2) DEFAULT 0 -- Average rating for display  
);
```

--MedicalRecord table for storing medical records for each Careseeker

```
CREATE TABLE MedicalRecord (  
    record_id INT PRIMARY KEY AUTO_INCREMENT,  
    careseeker_id INT,  
    record_type ENUM('Document', 'Image', 'Text') NOT NULL,  
    record_content TEXT,  
    upload_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (careseeker_id) REFERENCES Careseeker(careseeker_id) ON DELETE  
CASCADE  
);
```

--Consultant table with a foreign key reference to User

```
CREATE TABLE Consultant (  
    consultant_id INT PRIMARY KEY,  
    FOREIGN KEY (consultant_id) REFERENCES User(user_id) ON DELETE CASCADE,  
    expertise TEXT,  
    availability_status BOOLEAN,
```

```
rating DECIMAL(3, 2) DEFAULT 0,  
bio TEXT  
);
```

-- ApprovalRequest table for handling approval requests

```
CREATE TABLE ApprovalRequest (  
    request_id INT PRIMARY KEY AUTO_INCREMENT,  
    user_id INT, -- References the User table for both Caregivers and Consultants  
    request_type ENUM('Caregiver', 'Consultant') NOT NULL,  
    request_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    status ENUM('Pending', 'Approved', 'Declined') DEFAULT 'Pending',  
    comments TEXT,  
    FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE  
);
```

-- Document table for storing verification documents related to approval requests

```
CREATE TABLE Document (  
    document_id INT PRIMARY KEY AUTO_INCREMENT,  
    request_id INT, -- References the ApprovalRequest table  
    user_id INT, -- Redundant, but helps in directly associating document with a user  
    document_type VARCHAR(50),  
    file_path VARCHAR(255) NOT NULL,  
    upload_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (request_id) REFERENCES ApprovalRequest(request_id) ON DELETE  
    CASCADE,  
    FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE
```

```
);
```

-- Request table to handle service requests made by Careseekers to Caregivers or Consultants

```
CREATE TABLE Request (  
    request_id INT PRIMARY KEY AUTO_INCREMENT,  
    careseeker_id INT,  
    caregiver_id INT, -- Nullable, only filled for Caregiver requests  
    consultant_id INT, -- Nullable, only filled for Consultant requests  
    request_type ENUM('Caregiver', 'Consultant') NOT NULL, -- Type of request  
    request_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    status ENUM('Pending', 'Accepted', 'Cancelled') DEFAULT 'Pending',  
    comments TEXT,  
    FOREIGN KEY (careseeker_id) REFERENCES Careseeker(careseeker_id) ON DELETE CASCADE,  
    FOREIGN KEY (caregiver_id) REFERENCES Caregiver(caregiver_id) ON DELETE SET NULL,  
    FOREIGN KEY (consultant_id) REFERENCES Consultant(consultant_id) ON DELETE SET NULL  
);
```

--Appointment table for scheduling and tracking appointments

```
CREATE TABLE Appointment (  
    appointment_id INT PRIMARY KEY AUTO_INCREMENT,  
    request_id INT,  
    date DATE,  
    time TIME,  
    status ENUM('Scheduled', 'Completed', 'Cancelled') DEFAULT 'Scheduled',
```

```
notes TEXT,  
FOREIGN KEY (request_id) REFERENCES Request(request_id) ON DELETE CASCADE  
);
```

-- Payment table to handle payment information related to appointments

```
CREATE TABLE Payment (  
    payment_id INT PRIMARY KEY AUTO_INCREMENT,  
    appointment_id INT,  
    amount DECIMAL(10, 2),  
    payment_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    status ENUM('Paid', 'Unpaid', 'Pending') DEFAULT 'Pending',  
    FOREIGN KEY (appointment_id) REFERENCES Appointment(appointment_id) ON DELETE  
    CASCADE -- Payment is connect with Appointment when there is no advancement  
    payment and payment is only done after the service  
);
```

-- Review table to store reviews and ratings from users

```
CREATE TABLE Review (  
    review_id INT PRIMARY KEY AUTO_INCREMENT,  
    reviewer_id INT, -- User providing the review  
    reviewed_user_id INT, -- User receiving the review  
    review_role ENUM('Caregiver', 'Consultant', 'Careseeker') NOT NULL, -- Role of reviewed  
    user  
    rating DECIMAL(2, 1) CHECK (rating >= 1 AND rating <= 5), -- Rating out of 5  
    review_text TEXT,  
    review_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (reviewer_id) REFERENCES User(user_id) ON DELETE CASCADE,
```

```
FOREIGN KEY (reviewed_user_id) REFERENCES User(user_id) ON DELETE CASCADE  
);
```