**Patient CRUD Application Documentation**

**Overview**

This application is a basic PHP-based CRUD (Create, Read, Update, Delete) system for managing patient records in a MySQL database. The Patient class allows for the following operations:

* Create a new patient record
* Retrieve all or individual patient records
* Update an existing patient record
* Delete a patient record

**Prerequisites**

* PHP (version 7.4+ recommended)
* MySQL Database
* PDO extension for database operations

**Database Setup**

The application uses a database table named patients. Below is the suggested SQL schema for the patients table:

sql

Copy code

CREATE TABLE patients (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

age INT NOT NULL,

gender ENUM('Male', 'Female', 'Other') NOT NULL,

diagnosis TEXT,

date\_of\_birth DATE,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

**Class: Patient**

**Purpose**

The Patient class is responsible for interacting with the patients table in the database, providing methods to create, read, update, and delete records.

**Properties**

* **$conn** (PDO): The database connection object.
* **$table** (string): The name of the database table (patients).
* **$id** (int): The unique identifier for a patient.
* **$name** (string): The name of the patient.
* **$age** (int): The age of the patient.
* **$gender** (string): The gender of the patient, typically "Male," "Female," or "Other."
* **$diagnosis** (string): Diagnosis details or medical history of the patient.
* **$dateOfBirth** (string): The date of birth of the patient.

**Methods**

**\_\_construct()**

Initializes a new Patient object, establishing a database connection through the Database class.

* **Parameters**: None
* **Returns**: Void

**create()**

Creates a new patient record in the database.

* **Parameters**: None
* **Properties Used**: $name, $age, $gender, $diagnosis, $dateOfBirth
* **Returns**: bool - Returns true on successful execution, false on failure.
* **Example Usage**:

php

Copy code

$patient = new Patient();

$patient->name = "John Doe";

$patient->age = 35;

$patient->gender = "Male";

$patient->diagnosis = "Hypertension";

$patient->dateOfBirth = "1988-04-15";

$patient->create();

**fetchAll(int $limit = 5): array**

Fetches all patient records from the database, ordered by id in descending order. By default, it retrieves the latest 5 records.

* **Parameters**:
  + int $limit (optional): The number of records to fetch. Default is 5.
* **Returns**: array - An array of objects representing each patient.
* **Example Usage**:

php

Copy code

$patients = $patient->fetchAll();

foreach ($patients as $patient) {

echo $patient->name;

}

**fetchOne(): ?object**

Fetches a single patient record by id.

* **Parameters**: None
* **Properties Used**: $id
* **Returns**: object|null - Returns an object containing the patient details or null if not found.
* **Example Usage**:

php

Copy code

$patient = new Patient();

$patient->id = 1;

$patientRecord = $patient->fetchOne();

if ($patientRecord) {

echo $patientRecord->name;

}

**update()**

Updates an existing patient record in the database based on id.

* **Parameters**: None
* **Properties Used**: $id, $name, $age, $gender, $diagnosis, $dateOfBirth
* **Returns**: bool - Returns true on successful execution, false on failure.
* **Example Usage**:

php

Copy code

$patient = new Patient();

$patient->id = 1;

$patient->name = "Jane Doe";

$patient->age = 30;

$patient->gender = "Female";

$patient->diagnosis = "Diabetes";

$patient->dateOfBirth = "1993-07-23";

$patient->update();

**delete()**

Deletes a patient record from the database by id.

* **Parameters**: None
* **Properties Used**: $id
* **Returns**: bool - Returns true on successful execution, false on failure.
* **Example Usage**:

php

Copy code

$patient = new Patient();

$patient->id = 1;

$patient->delete();

**Pages**.

1). Create.php: This page allows the addition of a new patient which it does using the insert query.

2). Read.php: This page shows the list of all the patients in the system which is as a result of a select query.

3). Update.php: This page allows for the alter of a data for particular dataset in the db which is as the result of an Update query.

4). Delete.php: This page allows for the deletion of a particular dataset in the db which it does with a delete query.

**Classes**

1. Database: This is the class that houses the database connection. Shows also the db connection details like hostname, username, password e.t.c
2. Patient: This is the center of the CRUD application it houses the construnctor as well as all the CRUD functions like Create(), Fetchall(), update(), delete().

**SQL**

1. The db.sql contains the SQL queries to create the patients tables all well as the columns and their properties into the database.

**Error Handling**

It’s recommended to wrap database operations within try-catch blocks for better error handling and debugging.

php

Copy code

try {

$patient->create();

} catch (PDOException $e) {

echo "Error: " . $e->getMessage();

}

**Security Considerations**

* **Prepared Statements**: Used to protect against SQL injection by binding parameters in all SQL queries.
* **Input Validation**: Ensure that all user inputs are validated and sanitized before assignment to class properties.