**📄 README.md**

**🏥 Clinic Appointment Management System**

A Java Swing-based desktop application for managing clinic appointments, doctors, and patients. This project demonstrates core principles of **Java GUI development**, **database interaction via JDBC**, **MVC architecture**, **form validation**, and **report export functionality**.

**🔧 Technologies Used**

* **Java** (JDK 8+)
* **Java Swing** (for GUI)
* **JDBC** (Database connectivity)
* **MySQL** (Database)
* **MVC + DAO** design patterns
* Optional: **JavaMail API** for OTP login (in the RMI version)

**📁 Features**

**👤 Patient Management**

* Add, update, delete, and view patients
* Fields: Name, Email, Phone, Gender
* Input validation to ensure clean data entry

**👨‍⚕️ Doctor Management**

* Add, update, delete, and view doctors
* Fields: Name, Email, Specialty, Availability Status
* Enforces unique email and non-empty values

**📅 Appointment Booking**

* Schedule appointments between patients and doctors
* Prevents double booking or missing fields
* Buttons: Book, Update, Delete, View All

**🔐 Login System (Optional Upgrade)**

* Secure login with roles: admin, receptionist
* OTP login via email using JavaMail API
* "Forgot Password?" and show/hide password features

**📤 Export Features**

* Export appointment list to **CSV** or **Excel** (without external libraries)
* Data saved locally with the click of a button

**🗃️ Database Schema**

**📌 patients**

sql

CopyEdit

CREATE TABLE patients (

patient\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100),

email VARCHAR(100) UNIQUE,

phone VARCHAR(20),

gender VARCHAR(10)

);

**📌 doctors**

sql

CopyEdit

CREATE TABLE doctors (

doctor\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100),

email VARCHAR(100) UNIQUE,

specialization VARCHAR(100),

availability\_status VARCHAR(50)

);

**📌 appointments**

sql

CopyEdit

CREATE TABLE appointments (

appointment\_id INT AUTO\_INCREMENT PRIMARY KEY,

patient\_id INT,

doctor\_id INT,

appointment\_date DATE,

appointment\_time TIME,

FOREIGN KEY (patient\_id) REFERENCES patients(patient\_id),

FOREIGN KEY (doctor\_id) REFERENCES doctors(doctor\_id)

);

**📸 Screenshots (optional for GitHub)**

Include screenshots of your:

* Patient form
* Doctor form
* Appointment form
* Login interface (if available)

**🚀 How to Run**

1. Clone the project or download the .zip file
2. Set up your MySQL database using the schema above
3. Configure the JDBC connection in your DAO classes
4. Open the project in **NetBeans** or any Java IDE
5. Run Main.java to start the GUI

**🧠 Author**

**Musasira Kamali Philbert**

* Email: philbertkamali5@gmail.com
* Based in: Kigali, Rwanda
* Specialized in: Java, MySQL, Swing, Distributed Systems