**Project Name:**

Patient Health Information System

**Prepared By:**

Dingani F. Kandiwo, Joseph Boateng, and Khloe Edwards

**Date:**

11/27/2024

**Purpose:**

The purpose of this code is to manage operations within a hospital system, including employee access to sensitive data, patient diagnostics, insurance, and pharmacy management. This system is designed to streamline processes while maintaining data security and accuracy.

**Core Functionalities:**

**1. User Types and Authentication:**

* **Employee Access (E)**  
  Employees are authenticated using an ID and password stored in a JSON file. Based on their department, employees can access relevant functionalities:
  + Doctor (D)
  + Pharmacy (P)
  + Vitals (V)
  + Insurance (I)
* **Patient Access (P)**  
  Patients are verified using:
  + Security Questions: Verifies personal details like mother's maiden name, hospital of birth, and occupation.
  + OTP Verification: Sends a One-Time Password to the patient’s registered email.
* **New User Registration (new)**  
  Allows the addition of new employees in various departments:
  + Doctor, Pharmacy, Vitals, and Insurance.

**2. Functional Modules:**

**a. Doctor Module:**

Doctors can access patient health data stored in a JSON file and view detailed medical information.

**b. Pharmacy Module:**

Accesses and processes patient diagnosis results from a CSV file to retrieve drug and prescription data.

**c. Vitals Module:**

Displays patient diagnostic information based on their ID.

**d. Insurance Module:**

Retrieves and displays patient insurance information from a CSV file.

**e. Patient Module:**

Allows patients to access their reports via secure verification methods.

**3. Email Functionality:**

Generates and sends an OTP to the patient's registered email using SMTP.

**System Architecture:**

**Inputs:**

* Employee or Patient ID.
* Password for employee access.
* Security answers or OTP for patient verification.
* User-provided data during new user registration.

**Processes:**

* Authentication of users based on input credentials.
* Data retrieval from files (JSON and CSV formats).
* Validation of security answers or OTP.

**Outputs:**

* Access to patient information based on the user's department.
* OTPs sent to the patient’s email.
* Details such as diagnostics, prescriptions, or insurance data displayed.

**Error Handling:**

1. Invalid ID or Password:
   * Error messages for incorrect credentials.
2. File Read/Write Errors:
   * Fallback mechanisms in case of missing or corrupt files.
3. Invalid Inputs:
   * Clear prompts to guide users.

**Dependencies:**

* Python modules used:
  + json - For handling JSON files.
  + csv - For handling CSV files.
  + smtplib and email - For sending OTPs.
  + random - For generating OTPs.
* Data files:
  + employee.json - Stores employee credentials.
  + eye\_patient\_info.csv - Stores patient insurance data.
  + health\_data.json - Stores patient health data.
  + diagnosis\_results.csv - Stores diagnosis and pharmacy data.

**Constraints:**

* Requires pre-existing data files (JSON and CSV formats) with correct formatting.
* The SMTP server details and credentials must be correctly configured.

**Future Scope:**

1. Integration with a database to replace file-based data storage.
2. Implementation of advanced authentication mechanisms like biometrics.
3. Enhanced UI/UX for ease of access and interaction.
4. Addition of report generation for detailed analytics.