**Introduction:**

Hello everyone, my name is -- to the demo of my Meditech System application. In this video, I'll walk you through the setup process and demonstrate how to run the application successfully. Let's get started.

**Step 1: Setting Up the Environment:**

First, ensure you have Visual Studio Code installed on ymy system, along with the Python and MongoDB extensions.

**Step 2: Connecting to MongoDB:**

Open the Code.py file in Visual Studio Code and connect the MongoDB extension to ymy MongoDB application.

**Step 3: Importing Test Data:**

Next, import all the Meditech System JSON files into ymy MongoDB application. These files contain test data that I'll use to demonstrate the functionality of my application.

**Step 4: Configuring the Logo Path:**

In the Code.py file, locate the line where the logo image is loaded and resized. Update the logo\_path variable with the real path of ymy MediTech logo image. This ensures that the logo is displayed correctly in the application.

**Step 5: Running the Application:**

Once the setup is complete, you can run the Meditech System application. The application will start working seamlessly, allowing you to manage patients, doctors, appointments, invoices, staff, and medical records efficiently.

**Explaining the Code:**

Let's break down the main components and functionalities of the Meditech System code:

* **Importing Libraries:**
  + The code begins by importing necessary libraries such as tkinter, tkcalendar, Pillow, pymongo, and bson. These libraries provide functionalities for creating graphical user interfaces (GUI), handling images, interacting with MongoDB databases, and more.
* **Initializing MongoDB Client:**
  + The code initializes a MongoDB client using MongoClient from the pymongo library. It connects to the MongoDB server running on localhost with the default port 27017.
  + It also defines the database and collections needed for the application, such as users\_collection, patients\_collection, doctors\_collection, appointments\_collection, invoices\_collection, staff\_collection, and medical\_records\_collection.
* **Functions:**
* First one is Login, Logout function for the Proper Authentication. Then Patient,doctor,staff, appointment,invoice management systems that include multiple features like Add , delete, update the data as Ill there are search options, drop down list to select features are also been used for user to make the application easy to use.
* **Defining GUI Components:**
  + The code defines various GUI components using tkinter, such as labels, buttons, entry fields, and frames. These components are used to create different tabs and windows for different functionalities within the application.

**Demo:**

Let's now demonstrate the application's functionality. I will showcase each and every feature such as patient management, doctor management, appointment scheduling, invoice generation, staff management, and medical records management.

**Conclusion:**

That concludes my demo of the Meditech System application. I have shown you how to set up the environment, configure the application, and run it successfully. Thank you for watching, and I hope you find my application useful for managing healthcare operations efficiently.