Pneumonia Detection System – Functional Overview

1. model.py

This file handles loading and verification of the deep learning model used for pneumonia detection. It ensures that the trained model file exists, logs details about it, and loads it for prediction. If the model is missing or fails to load, a user-friendly error is shown.

- **check_model_path()**: Verifies the existence and size of the model file (model.h5) and logs essential details.

- **get_model()**: Loads the TensorFlow/Keras model into memory for prediction. Displays an error if loading fails.

2. signup.py

This file defines the **SignUpWindow** class, which manages the registration (sign-up) process for new users. It creates a graphical interface using Tkinter where users can enter their email and password.

- **__init__()**: Builds the sign-up page layout, sets background images, and designs input fields for email and password.

- **sign_up()**: Validates inputs, hashes the password, and saves user credentials into the SQLite database.

- **go_to_login()**: Switches from the sign-up screen to the login screen after successful registration.

3. login.py

The **LoginWindow** class provides the user login interface. It verifies user credentials stored in the database and opens the pneumonia detection interface upon successful authentication.

- **__init__()**: Builds the login page layout, including input fields and navigation buttons.

- **login()**: Checks the entered email and password against database records using bcrypt verification.

- **go_to_signup()**: Allows users to switch to the sign-up window if they don’t have an account.

4. detection.py

This is the core functionality of the system — the **DetectionWindow** class handles image upload, prediction, and report generation. It also records test history into the database.

- **upload_image()**: Allows users to select a chest X-ray image and preview it in the interface. Checks for grayscale format.

- **preprocess_image()**: Resizes and normalizes the uploaded image to prepare it for model prediction.

- **predict()**: Uses the pre-trained model to classify the X-ray as either 'Normal' or 'Pneumonia'. Saves results to the database.

- **regenerate_report()**: Generates a PDF report summarizing the prediction result and saves it in a 'Reports' folder.

- **log_out()**: Safely logs the user out and redirects back to the login screen.

5. database.py

This file handles all database-related operations, including table creation and password security.

- **init_db()**: Creates the required tables (`users` and `tests`) if they don’t already exist.

- **hash_password()**: Hashes user passwords using bcrypt before storing them for enhanced security.

- **verify_password()**: Verifies entered passwords against stored hashed passwords during login.

6. Overall Workflow

1. The user signs up or logs in using their credentials. 2. Once logged in, the user uploads a chest X-ray image. 3. The system preprocesses the image and uses the trained model to detect pneumonia. 4. The result is displayed and saved in the database. 5. The user can generate a downloadable PDF report. 6. The user can log out and return to the login page.

End of Report