PROJECT DOCUMENTATION

DATABASE FOR PHARMACY CHAIN

I. The project envisions the creation of a database for a pharmacy network and an application aimed at managing information about medications, patients, prescriptions, and employees of individual pharmacies.

The database should enable:

- Browse available medications at the pharmacy
- Employee and patient data management
- Recording filled prescriptions
- Generating reports (useful, e.g. for stocktaking)

The main functions of the application include:

- Adding and updating individual entries as well as deleting outdated data
- The ability to view data in a convenient and intuitive manner
- Storing the history of prescriptions filled by the pharmacies
- Creating summaries (reports)

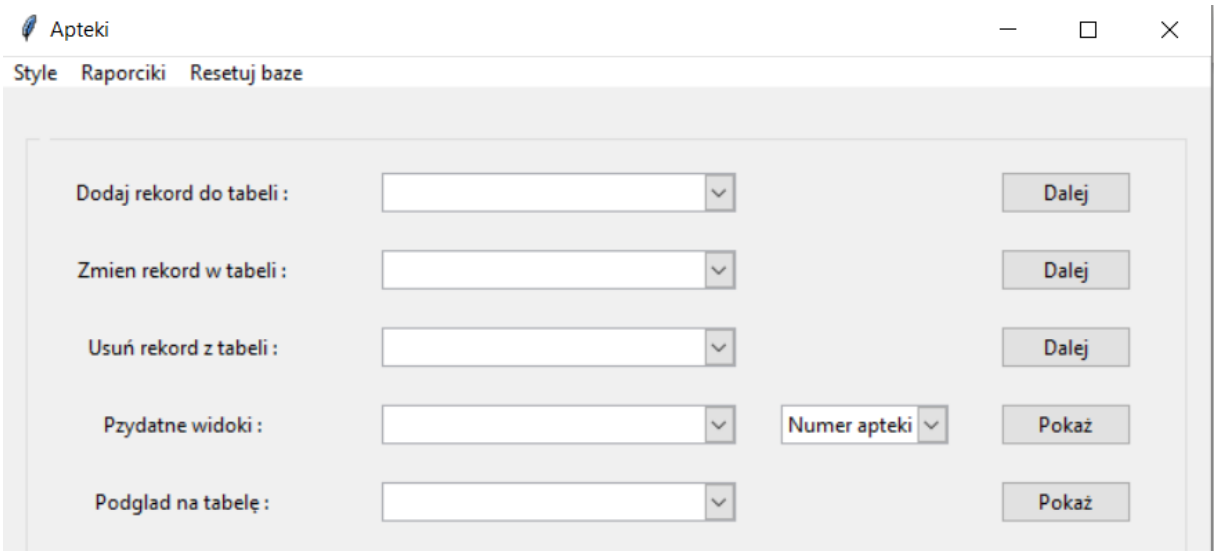
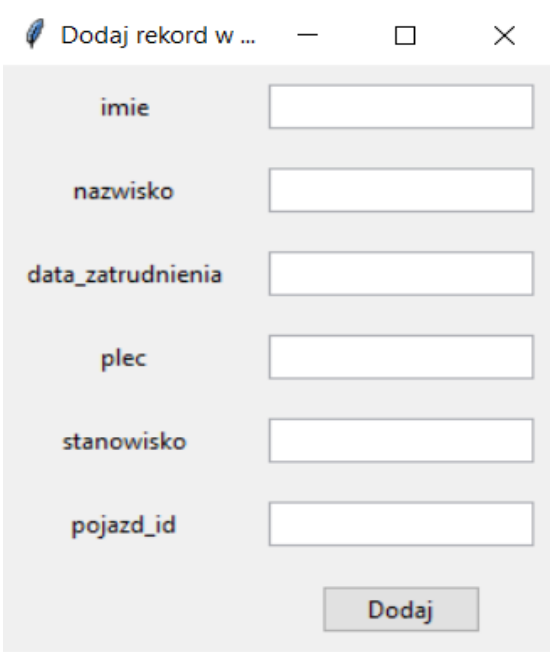
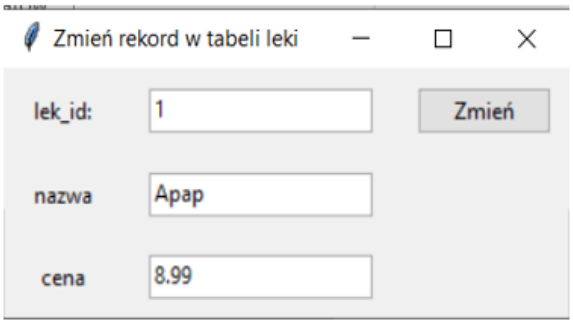


Fig. 1: Main application control panel (select the table where you want to make a specific change or view, then click the 'next' button (in the case of adding, deleting, or modifying a record in the table - you must provide the record ID or new data) or the 'show' button (in the case of views or simply viewing the table)).





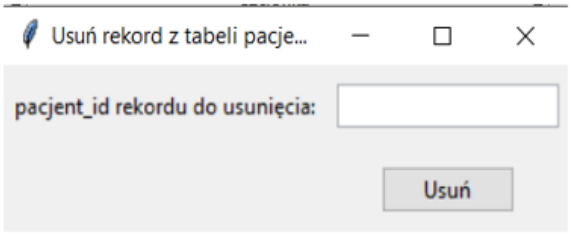


Fig. 2,3,4: Example windows after selecting a table and clicking the 'next' button.

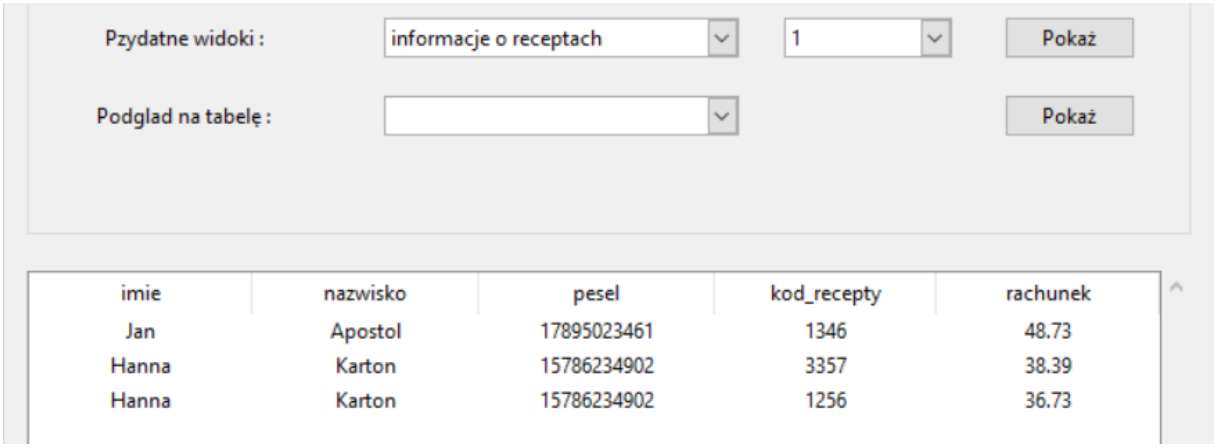


Fig. 5 : Example view for pharmacy nb 1

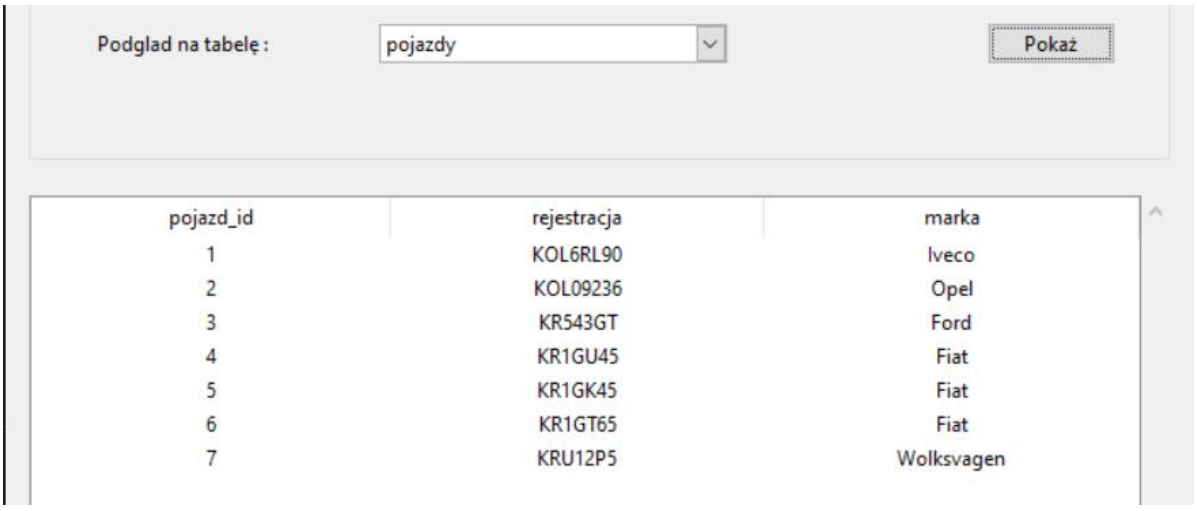


Fig.6 : Example table preview

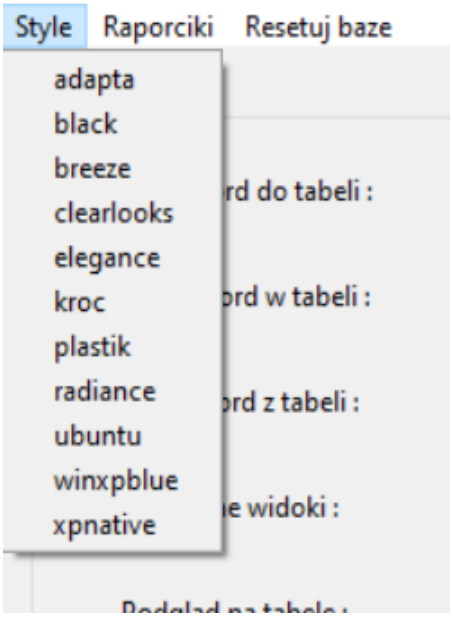


Fig. 7: Changing the application's appearance styles

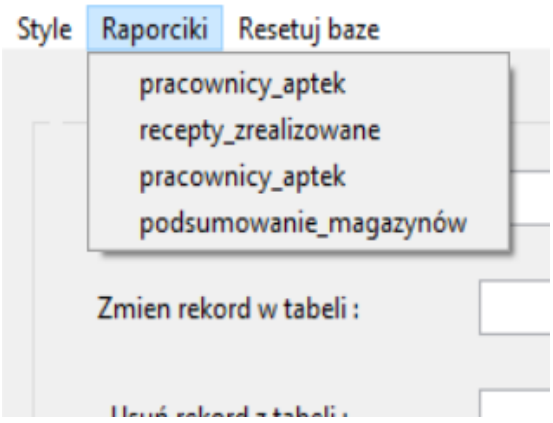
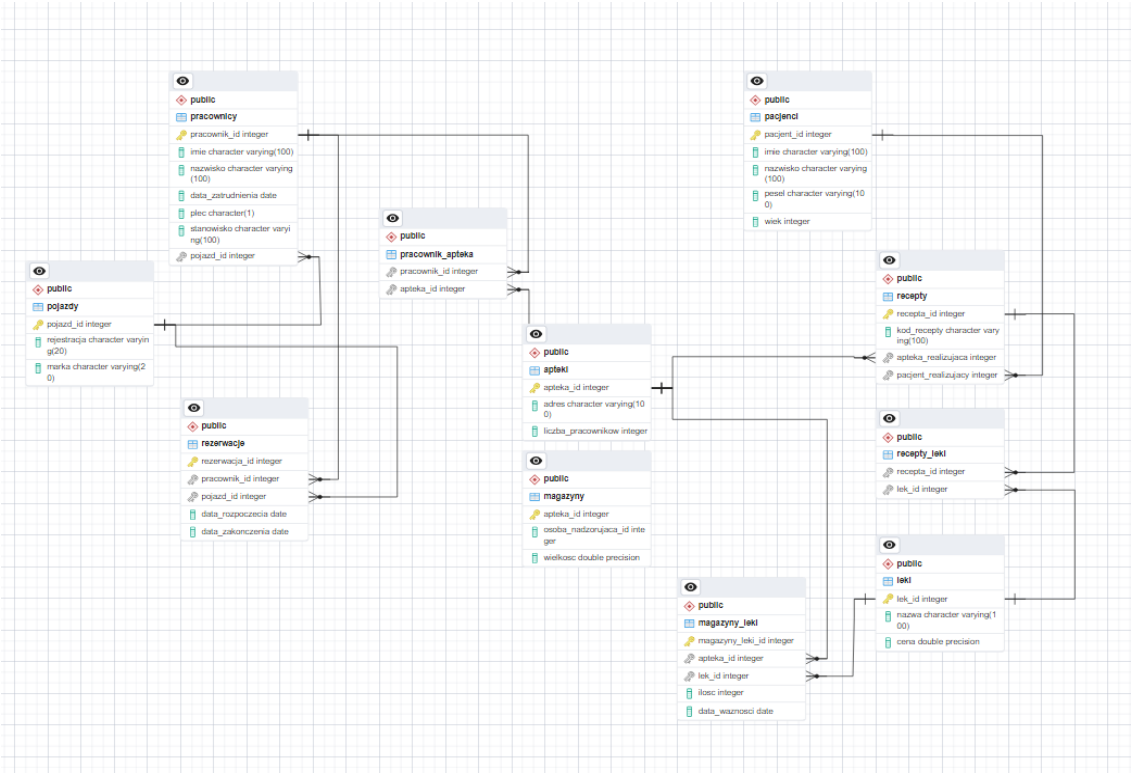


Fig. 8: Buttons for generating reports (next to it, a reset button)

II. The sentence you provided translates to: "The logic of my database defined in the ERD below is generated by the pgAdmin tool."



III. The interface for presenting, editing, and handling data is unified (there is no division into users). Data entry and deletion into tables are carried out by selecting the appropriate table and filling in the values of all columns in a separate window. Editing data is similar, but first, we select the record to edit by entering the ID of that record from the given table.
Data visualization is implemented in two ways:

- Quick preview of individual tables.
- Viewing selected views for individual pharmacies.

Views are created to facilitate the user in gaining a broader perspective on tables related to each other.

The created views include:

- Warehouse summary - contains information about the person supervising the warehouse, the name of the medication, its price, and the quantity in the warehouse.

- Employee summary - the first and last name of the employee, gender, date of employment, and their position.
- Prescription summary - the first and last name of the patient, personal ID number, prescription code, and bill (as the sum of the prices of all medications).
- Reservation summary - the first and last name of the employee, position, brand, and registration of the reserved car, and the start and end date of the reservation.

Reports will be generated by the application after clicking the"Reports" tab and selecting the appropriate report. Reports contain information such as views and are generated into a text file in the form: each line contains the values of individual columns. These values are grouped according to pharmacy numbers. An additional tab allows for changing the style of the application's appearance.

IV. Data can be entered into the database individually from within the application. By clicking the 'reset database' tab and confirming the reset, the database will be reset to its initial state (by loading the 'reset.sql' file). Additionally, I am including the 'queries.sql' file with all the SQL queries I used during the project creation.