**To-Do Desktop Application**

**Overview:**

The TODO Desktop Application is designed to help users keep track of their daily tasks. It is implemented using PyCharm as the development environment, SQLite as the database handler, and QT Designer for designing the graphical user interface (GUI). It is developed using the Python programming language and various libraries, including SQLite, and QT Designer by importing them into our working environment. This application allows users to add tasks with deadlines, view tasks in a table format, delete the tasks which are done and view details about a task by clicking on it.

**Dependencies:**

**GUI Design:**

The graphical user interface (GUI) of the app was designed using QT Designer. The design includes several widgets, including a calendar widget for selecting task due dates, time widget for selecting task due time, line edit widgets for entering task titles and descriptions, a table widget for displaying task information, and push button widgets for adding, listing, and viewing tasks. QT Designer is a tool that allows you to visually design GUIs by dragging and dropping widgets. Once you have designed your GUI, you can convert it to Python code using the QT Designer library. The GUI of the app is designed using the Qt Designer tool, which generates a .ui file. The .ui file is converted to Python code using the pyuic5 tool, which generates a module that can be imported into the main Python script.

**SQLite Database:**

SQLite is a lightweight and fast relational database system, which is well-suited for small to medium-sized projects like a to-do app. And by creating a separate Python file that imports the sqlite3 module and interacts with the database, you've kept your code organized and modular. The database consists of a single table with columns for the task title, description, due date, and due time. When a user adds a task, the task information is inserted into the database using SQL queries. When a user lists tasks, the necessary information is retrieved from the database using SQL queries.

**PyCharm:**

An integrated development environment (IDE) used for writing, debugging and testing the Python code. The Python code for your app consists of several files, including:

* **main.py:** This file contains the Python code generated from your QtDesigner design. It imports the necessary Qt libraries and sets up the main window for your app.
* **DBHandler.py:** This file contains Python code for interacting with the SQLite database. It imports the sqlite3 library and defines functions for adding, listing, deleting, and viewing tasks.

**Installation:**

To run this application, please follow the steps below:

1. Install PyCharm on your desktop.
2. Clone the repository from GitHub.
3. Open the project in PyCharm.
4. Install the required libraries using pip (PyQt5, sqlite3)

Run the main.py file to start the application when the user adds a task, the insertTask() function in DBHandler.py is called. This function retrieves the task information from the GUI widgets and inserts it into the database using SQL queries. When the user clicks the "List Tasks" button, all tasks from the database that are due on the current date and populates the table widget in the GUI with the task information. When the user clicks on a task in the table widget, the getTask() function in DBHandler.py is called. This function retrieves the task information from the database and displayed.

**Features:**

* **Add Task Button:** Allows users to add a new task by providing a title, description, deadline date and time.
* **List Task Button:** This button allows users to view their tasks in a table format. The table displays the task ID, title, description, deadline date, and time. Users can click on a task to view its title and description by.
* **View Task Button:** This button allows users to view the title and description of a task selected from the table.
* **Delete Task Button:** This button allows users to delete the selected task which is done.

**Conclusion:**

By using PyCharm, SQLite, and QT Designer, I have created a TODO Desktop Application with a clean GUI and allowing users to add tasks to the app by selecting a date from a calendar, setting a deadline time, and providing a title and description is a great way to make task entry easy. And being able to view all the day's tasks in a table and delete tasks, and view details about a task by clicking on it. click on a task to view title and description is a useful feature.