Simple GUI for Task Management

AIM:

The aim of the "Task Management GUI" project is to create a simple and user-friendly application that allows users to easily add, view, and delete tasks. This helps users organize their tasks efficiently and keep track of what they need to do. The project demonstrates basic programming and GUI design skills in MATLAB.

THEORY:

User Interface (UI):

A GUI provides a visual way for users to interact with the software. Instead of typing commands, users can click buttons and enter text, making it more accessible.

Task Management:

The project focuses on helping users manage their tasks efficiently. Users can add tasks they need to complete and delete tasks they have finished, promoting organization.

Data Handling:

Tasks are stored in a list (array) in memory. This allows the program to dynamically update the displayed list whenever tasks are added or removed.

Event-Driven Programming:

The application operates on user events, such as button clicks. When a button is pressed, specific functions are triggered to perform actions like adding or deleting tasks.

MATLAB GUI Components:

The project uses various GUI components like text boxes for input, buttons for actions, and list boxes for displaying tasks. Understanding these components is crucial for building interactive applications.

Feedback and Updates:

The interface provides immediate feedback. For example, when a task is added, the list updates instantly, allowing users to see their changes right away.

Programming Skills:

Building this project helps improve skills in MATLAB, particularly in areas like GUI design, managing data structures, and writing callback functions for user interactions.

Potential Enhancements:

The project can be expanded to include features like task prioritization, deadlines, or saving tasks to a file, making it a more robust task management tool.

Real-World Application:

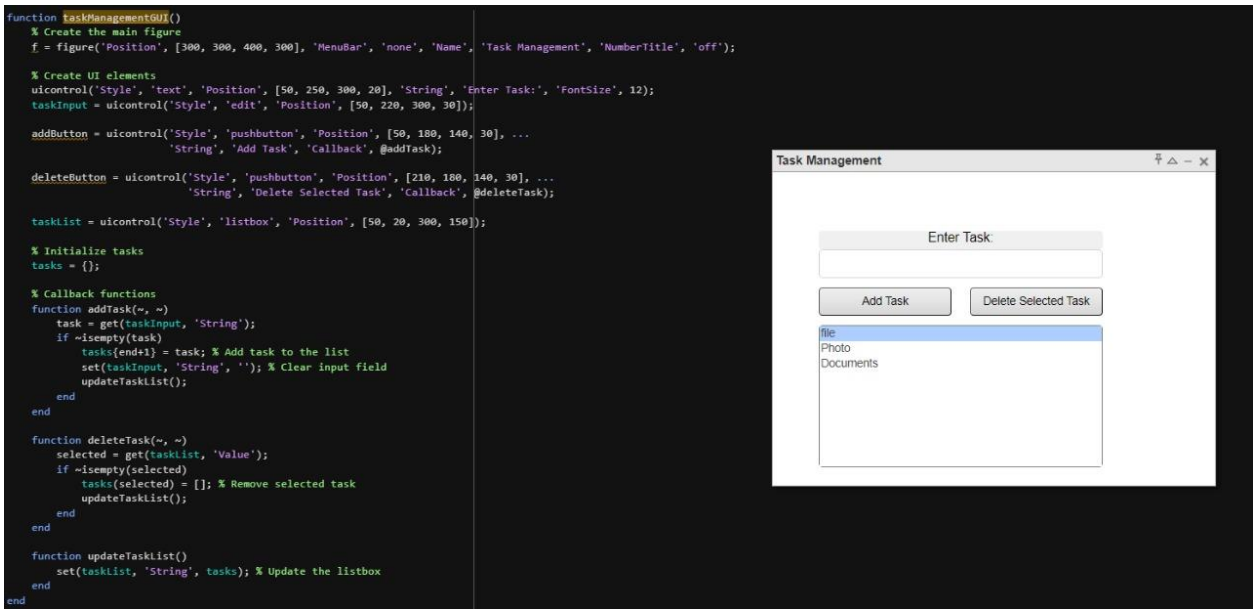
Task management is a common need in both personal and professional settings, making this project relevant and useful for everyday life.

PROGRAM:

```
function taskManagementGUI()
    % Create the main figure
    f = figure('Position', [300, 300, 400, 300], 'MenuBar', 'none', 'Name', 'Task Management', 'NumberTitle', 'off');
    % Create UI elements
    uicontrol('Style', 'text', 'Position', [50, 250, 300, 20], 'String', 'Enter Task:', 'FontSize', 12);
    taskInput = uicontrol('Style', 'edit', 'Position', [50, 220, 300, 30]);
    addButton = uicontrol('Style', 'pushbutton', 'Position', [50, 180, 140, 30], ...
                          'String', 'Add Task', 'Callback', @addTask);
    deleteButton = uicontrol('Style', 'pushbutton', 'Position', [210, 180, 140, 30], ...
                             'String', 'Delete Selected Task', 'Callback', @deleteTask);
    taskList = uicontrol('Style', 'listbox', 'Position', [50, 20, 300, 150]);
    % Initialize tasks
    tasks = {};
    % Callback functions
    function addTask(~, ~)
        task = get(taskInput, 'String');
        if ~isempty(task)
            tasks{end+1} = task; % Add task to the list
            set(taskInput, 'String', ''); % Clear input field
            updateTaskList();
        end
    end
```

```
    function deleteTask(~, ~)
        selected = get(taskList, 'Value');
        if ~isempty(selected)
            tasks(selected) = []; % Remove selected task
            updateTaskList();
        end
    end
    function updateTaskList()
        set(taskList, 'String', tasks); % Update the listbox
    end
end
```

OUTPUT:



RESULT:

The project successfully creates a user-friendly application that allows users to easily add, view, and delete tasks, helping them stay organized and focused on their work. The simple design ensures accessibility for all users, making task management efficient and effective.