

- Develop a simple To-Do List application using Java.

## **Java Code:**

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;

public class TodoListSwing extends JFrame {
    private static ArrayList<String> tasks = new ArrayList<>();
    private DefaultListModel<String> tasksListModel;
    private JList<String> tasksList;
    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new
        TodoListSwing().setVisible(true));
    }
    public TodoListSwing() {
        this.tasksListModel = new DefaultListModel<>();
        this.tasksList = new JList<>(tasksListModel);
        JButton addButton = new JButton("Add Task");
```

```
addButton.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        addTask();  
    }  
});  
  
JButton deleteButton = new JButton("Delete Task");  
deleteButton.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        deleteTask();  
    }  
});  
  
JButton displayButton = new JButton("Display Tasks");  
displayButton.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        displayTasks();  
    }  
});  
  
JButton completeButton = new JButton("Mark Task as  
Complete");  
completeButton.addActionListener(new ActionListener() {  
    @Override
```

```

        public void actionPerformed(ActionEvent e) {
            markAsComplete();
        }
    });

    JPanel buttonPanel = new JPanel();
    buttonPanel.setLayout(new FlowLayout());
    buttonPanel.add(addButton);
    buttonPanel.add(deleteButton);
    buttonPanel.add(displayButton);
    buttonPanel.add(completeButton);

    JPanel mainPanel = new JPanel();
    mainPanel.setLayout(new BorderLayout());
    mainPanel.add(new JScrollPane(tasksList),
BorderLayout.CENTER);

    mainPanel.add(buttonPanel, BorderLayout.SOUTH);
    add(mainPanel);
    setTitle("Todo List Application");
    setSize(400, 300);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLocationRelativeTo(null);
}

private void addTask() {
    String task = JOptionPane.showInputDialog(this, "Enter the
task:");

```

```
if (task != null && !task.trim().isEmpty()) {
    tasks.add(task);
    tasksListModel.addElement(task);
    JOptionPane.showMessageDialog(this, "Task added
successfully!");
} else {
    JOptionPane.showMessageDialog(this, "Task cannot be
empty!");
}
}

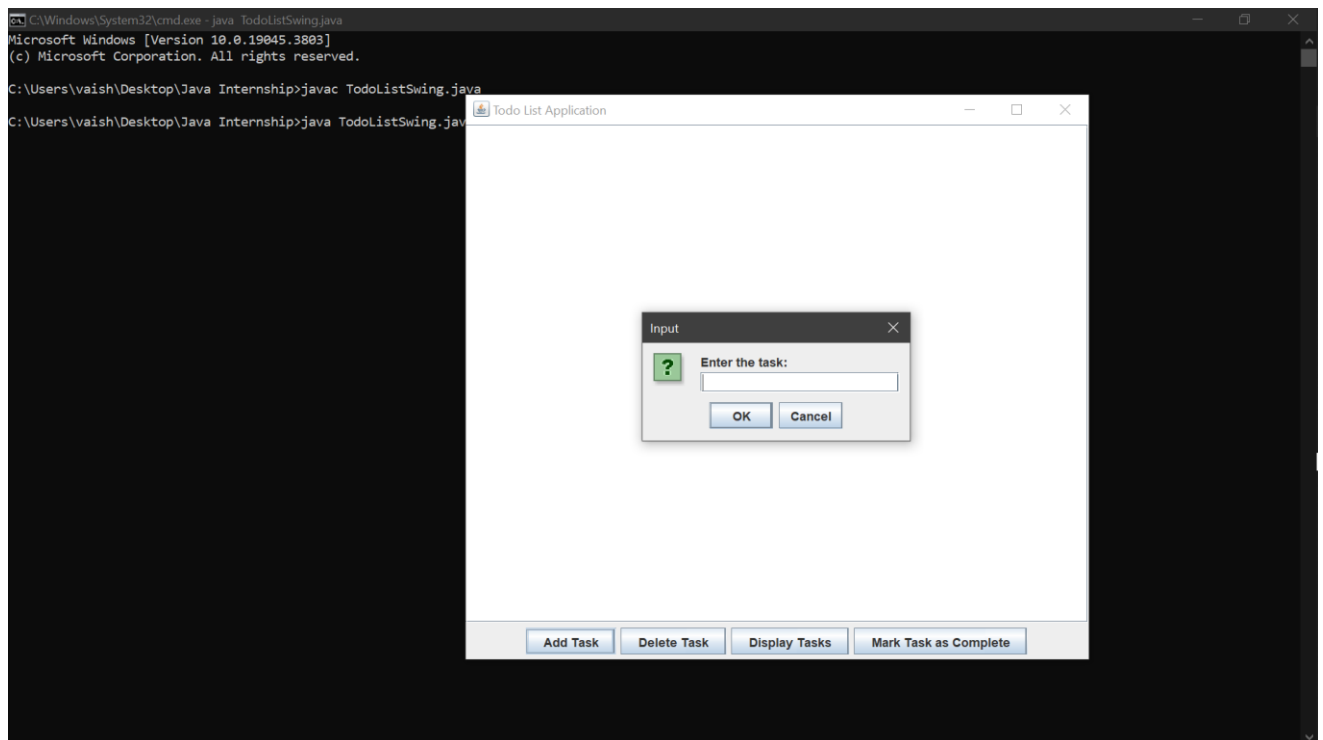
private void deleteTask() {
    if (tasks.isEmpty()) {
        JOptionPane.showMessageDialog(this, "No tasks in the list.");
        return;
    }
    int index = tasksList.getSelectedIndex();
    if (index >= 0 && index < tasks.size()) {
        tasks.remove(index);
        tasksListModel.removeElementAt(index);
        JOptionPane.showMessageDialog(this, "Task deleted
successfully!");
    } else {
        JOptionPane.showMessageDialog(this, "Please select a task to
delete.");
    }
}
```

```
}
```

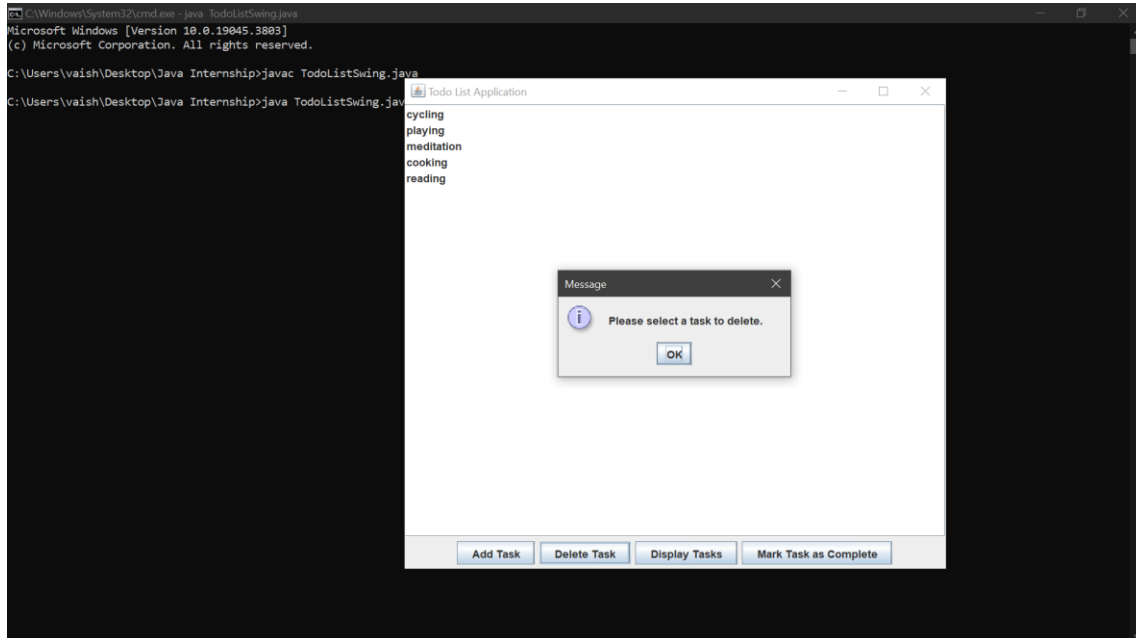
```
private void displayTasks() {  
    if (tasks.isEmpty()) {  
        JOptionPane.showMessageDialog(this, "No tasks in the list.");  
    } else {  
        StringBuilder tasksText = new StringBuilder("Tasks:\n");  
        for (int i = 0; i < tasks.size(); i++) {  
            tasksText.append(i).append(".  
").append(tasks.get(i)).append("\n");  
        }  
        JOptionPane.showMessageDialog(this, tasksText.toString());  
    }  
}  
  
private void markAsComplete() {  
    if (tasks.isEmpty()) {  
        JOptionPane.showMessageDialog(this, "No tasks in the list.");  
        return;  
    }  
  
    int index = tasksList.getSelectedIndex();  
    if (index >= 0 && index < tasks.size()) {  
        String task = tasks.get(index);  
        tasks.set(index, "[Completed] " + task);  
        tasksListModel.setElementAt("[Completed] " + task, index);  
    }  
}
```

```
        JOptionPane.showMessageDialog(this, "Task marked as
complete!");
    } else {
        JOptionPane.showMessageDialog(this, "Please select a task to
mark as complete.");
    }
}
}
```

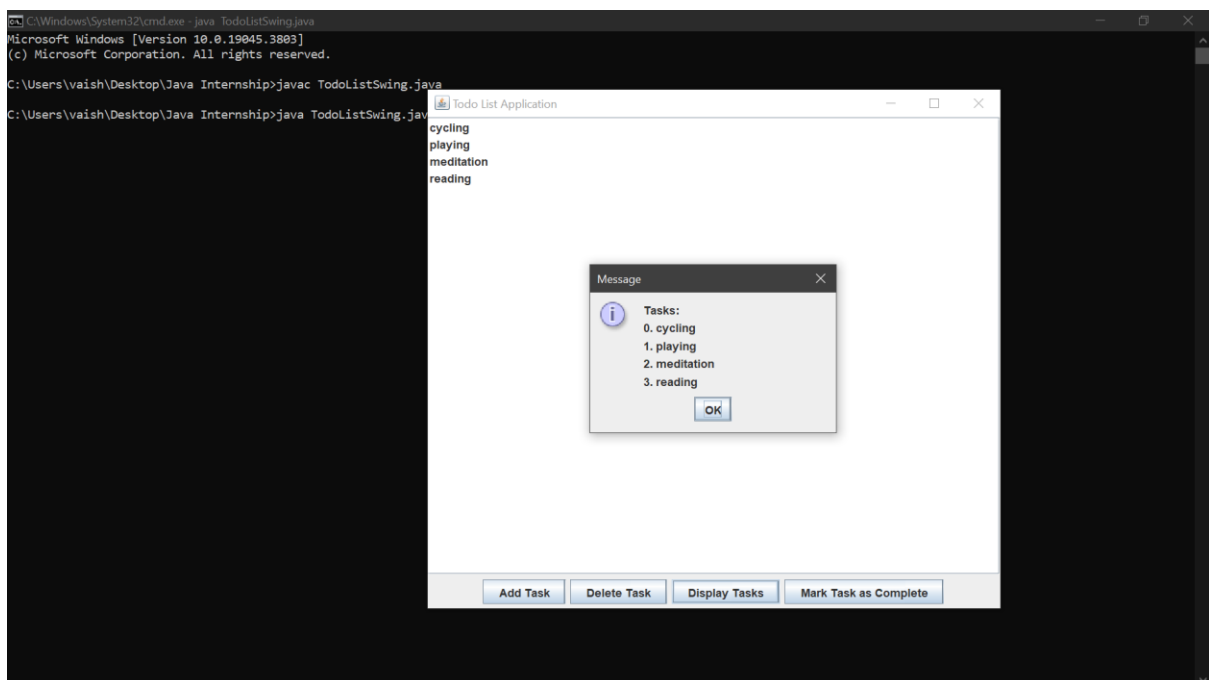
## Image while Adding task :



## Image while Deleting task :



## Image while Displaying task :



## Image while Mark task as completed :

