```
function microgrid_dashboard()

    fig = uifigure('Name', 'Microgrid Load Flow and Fault Analysis Dashboard', ...
                   'Position', [100, 100, 1400, 900], ...
                   'Color', [0.94, 0.94, 0.94]);


    tabgroup = uitabgroup(fig, 'Position', [10, 10, 1380, 880]);

    % Tab 1: System Overview
    tab1 = uitab(tabgroup, 'Title', 'System Overview');
    create_system_overview(tab1);

    % Tab 2: Load Flow Analysis
    tab2 = uitab(tabgroup, 'Title', 'Load Flow Analysis');
    create_load_flow_tab(tab2);

    % Tab 3: Fault Analysis
    tab3 = uitab(tabgroup, 'Title', 'Short Circuit Analysis');
    create_fault_analysis_tab(tab3);

    % Tab 4: Real-time Monitoring
    tab4 = uitab(tabgroup, 'Title', 'Real-time Monitoring');
    create_monitoring_tab(tab4);
end

function create_system_overview(parent)

    title_label = uilabel(parent, 'Text', 'Microgrid System Overview', ...
                          'Position', [20, 820, 300, 30], ...
                          'FontSize', 18, 'FontWeight', 'bold');


    specs_panel = uipanel(parent, 'Title', 'System Specifications', ...
                          'Position', [20, 500, 400, 300], ...
                          'FontSize', 12, 'FontWeight', 'bold');


    specs_text = {
        'Utility Specifications:'
        '• Nominal Voltage: 230 kV'
        '• 3LG Fault Current: 5000 A (X/R = 10)'
        '• SLG Fault Current: 7000 A (X/R = 12)'
        ''
        'Transformer (T1):'
        '• Rating: 100 MVA'
        '• Primary: 230 kV'
        '• Secondary: 13.8 kV'
        '• Impedance: 10%'
        '• Connection: Dyn1'
        ''
```

```
        'Underground Cable:'
        '• Type: 15MALS1'
        '• Size: 750 MCM'
        '• Length: 0.1 miles'
        '• Conductors per phase: 12'
        ''
        'Feeder Loads:'
        '• Feeder 1 & 2: 40 MVA each'
        '• Power Factor: 0.8 lagging'
        '• Connection: Delta'
    };

    uitextarea(specs_panel, 'Value', specs_text, ...
              'Position', [10, 10, 370, 270], ...
              'Editable', 'off');

    % Single-line diagram
    diagram_panel = uipanel(parent, 'Title', 'Single Line Diagram', ...
                            'Position', [450, 400, 900, 400], ...
                            'FontSize', 12, 'FontWeight', 'bold');


    ax = uiaxes(diagram_panel, 'Position', [20, 20, 860, 360]);
    draw_single_line_diagram(ax);

    bus_panel = uipanel(parent, 'Title', 'Bus Data Summary', ...
                        'Position', [20, 50, 1330, 330], ...
                        'FontSize', 12, 'FontWeight', 'bold');

    bus_data = {
        'T_sub', '230.0', '100.0', '0.0', '62.88', '53.87', 'Swing Bus'
        'T1_HS', '230.0', '100.0', '0.0', '-62.88', '-53.88', 'Load Bus'
        'T1_LS', '13.8', '94.6', '-3.7', '62.68', '47.03', 'Load Bus'
        'SUBSTATION BUS', '13.8', '94.6', '-3.7', '-62.64', '-46.98', 'Load Bus'
    };

    bus_table = uitable(bus_panel, 'Data', bus_data, ...
                        'ColumnName', {'Bus ID', 'Nominal kV', 'Voltage %', 'Angle °',↙
'MW', 'Mvar', 'Type'}, ...
                        'Position', [20, 20, 1290, 290]);
end

function create_load_flow_tab(parent)

    title_label = uilabel(parent, 'Text', 'Load Flow Analysis Results', ...
                          'Position', [20, 820, 300, 30], ...
                          'FontSize', 18, 'FontWeight', 'bold');

    voltage_panel = uipanel(parent, 'Title', 'Bus Voltage Profile', ...
                            'Position', [20, 550, 650, 250], ...
                            'FontSize', 12, 'FontWeight', 'bold');
```

```
    ax1 = uiaxes(voltage_panel, 'Position', [30, 30, 600, 200]);

    bus_names = {'T_sub', 'T1_HS', 'T1_LS', 'SUBSTATION BUS'};
    voltage_pu = [1.00, 1.00, 0.946, 0.946];
    voltage_angle = [0.0, 0.0, -3.7, -3.7];

    bar(ax1, voltage_pu);
    ax1.XTickLabel = bus_names;
    ax1.Title.String = 'Bus Voltage Magnitude (p.u.)';
    ax1.YLabel.String = 'Voltage (p.u.)';
    grid(ax1, 'on');

    power_panel = uipanel(parent, 'Title', 'Power Flow Distribution', ...
                          'Position', [690, 550, 650, 250], ...
                          'FontSize', 12, 'FontWeight', 'bold');

    ax2 = uiaxes(power_panel, 'Position', [30, 30, 600, 200]);

    mw_data = [62.88, -62.88, 62.68, -62.64];
    mvar_data = [53.87, -53.88, 47.03, -46.98];

    hold(ax2, 'on');
    bar(ax2, [mw_data', mvar_data']);
    ax2.XTickLabel = bus_names;
    ax2.Title.String = 'Active and Reactive Power Flow';
    ax2.YLabel.String = 'Power (MW/Mvar)';
    legend(ax2, 'MW', 'Mvar');
    grid(ax2, 'on');

    branch_panel = uipanel(parent, 'Title', 'Branch Loading Summary', ...
                           'Position', [20, 280, 1320, 250], ...
                           'FontSize', 12, 'FontWeight', 'bold');

    branch_data = {
        'T1 (Transformer)', '100 MVA', '82.81 MVA', '82.8%', '166.7 A', 'Normal'
        'UG CABLE', '6144 A', '3465 A', '56.4%', '3465 A', 'Normal'
        'Line1 (Transmission)', '—', '—', '—', '207.9 A', 'Normal'
    };

    branch_table = uitable(branch_panel, 'Data', branch_data, ...
                           'ColumnName', {'Branch', 'Capacity', 'Loading', 'Loading %',↙
'Current', 'Status'}, ...
                           'Position', [20, 20, 1280, 210]);

    summary_panel = uipanel(parent, 'Title', 'System Summary', ...
                            'Position', [20, 50, 1320, 210], ...
                            'FontSize', 12, 'FontWeight', 'bold');

    summary_text = {
        'Load Flow Convergence: 4 iterations'
```

```
        'System Mismatch: 0.000 MW, 0.000 Mvar'
        'Total Generation: 62.88 MW, 53.87 Mvar'
        'Total Load: 62.88 MW, 53.87 Mvar'
        'Total Losses: 0.237 MW, 6.884 Mvar'
        'System Power Factor: 75.9% Lagging'
        ''
        'Alerts:'
        '● SUBSTATION BUS: Under Voltage (94.6% < 95.0%)'
        '● T1_LS: Under Voltage (94.6% < 95.0%)'
    };

    uitextarea(summary_panel, 'Value', summary_text, ...
              'Position', [20, 20, 1280, 170], ...
              'Editable', 'off');
end

function create_fault_analysis_tab(parent)

    title_label = uilabel(parent, 'Text', 'Short Circuit Analysis Results', ...
                          'Position', [20, 820, 300, 30], ...
                          'FontSize', 18, 'FontWeight', 'bold');

    fault_panel = uipanel(parent, 'Title', 'Fault Current Analysis', ...
                          'Position', [20, 550, 650, 250], ...
                          'FontSize', 12, 'FontWeight', 'bold');

    ax1 = uiaxes(fault_panel, 'Position', [30, 30, 600, 200]);

    fault_types = {'3-Phase', 'Line-Ground', 'Line-Line', 'Line-Line-Ground'};
    fault_currents = [45.022, 43.658, 38.990, 44.744]; % kA

    bar(ax1, fault_currents);
    ax1.XTickLabel = fault_types;
    ax1.Title.String = 'Fault Currents at SUBSTATION BUS';
    ax1.YLabel.String = 'Fault Current (kA)';
    grid(ax1, 'on');

    impedance_panel = uipanel(parent, 'Title', 'Sequence Impedances', ...
                              'Position', [690, 550, 650, 250], ...
                              'FontSize', 12, 'FontWeight', 'bold');

    ax2 = uiaxes(impedance_panel, 'Position', [30, 30, 600, 200]);

    seq_types = {'Positive', 'Negative', 'Zero'};
    resistance = [0.01287, 0.01287, 0.00846];
    reactance = [0.17650, 0.17650, 0.19341];

    hold(ax2, 'on');
    bar(ax2, [resistance', reactance']);
    ax2.XTickLabel = seq_types;
    ax2.Title.String = 'Sequence Impedances (Ohms)';
```

```
    ax2.YLabel.String = 'Impedance (Ohms)';
    legend(ax2, 'Resistance', 'Reactance');
    grid(ax2, 'on');

    % Fault contribution table
    contrib_panel = uipanel(parent, 'Title', 'Fault Current Contributions', ...
                            'Position', [20, 280, 1320, 250], ...
                            'FontSize', 12, 'FontWeight', 'bold');

    contrib_data = {
        'Total System', '45.022', '43.658', '38.990', '44.744'
        'T1 Transformer', '27.714', '32.468', '—', '—'
        'Feeder 1 Load', '8.659', '5.598', '—', '—'
        'Feeder 2 Load', '8.659', '5.598', '—', '—'
    };

    contrib_table = uitable(contrib_panel, 'Data', contrib_data, ...
                            'ColumnName', {'Source', '3-Phase (kA)', 'L-G (kA)', 'L-L↙
(kA)', 'L-L-G (kA)'}, ...
                            'Position', [20, 20, 1280, 210]);

    % Fault analysis summary
    fault_summary_panel = uipanel(parent, 'Title', 'Analysis Summary', ...
                                  'Position', [20, 50, 1320, 210], ...
                                  'FontSize', 12, 'FontWeight', 'bold');

    fault_summary_text = {
        'Fault Analysis at SUBSTATION BUS (13.8 kV):'
        '• Pre-fault Voltage: 100% of nominal'
        '• Maximum Fault Current: 45.022 kA (3-Phase Fault)'
        '• Minimum Fault Current: 38.990 kA (Line-Line Fault)'
        ''
        'System Strength:'
        '• Utility Short Circuit Power: 1991.858 MVA'
        '• System X/R Ratio: 34.1 (Transformer), 10.0 (Utility)'
        ''
        'Protection Considerations:'
        '• Circuit breakers must be rated for minimum 45 kA'
        '• Coordination required between utility and feeder protection'
        '• Ground fault protection needed for Dyn transformer connection'
    };

    uitextarea(fault_summary_panel, 'Value', fault_summary_text, ...
              'Position', [20, 20, 1280, 170], ...
              'Editable', 'off');
end

function create_monitoring_tab(parent)

    title_label = uilabel(parent, 'Text', 'Real-time System Monitoring', ...
                          'Position', [20, 820, 300, 30], ...
```

```
                    'FontSize', 18, 'FontWeight', 'bold');

    control_panel = uipanel(parent, 'Title', 'Control Panel', ...
                        'Position', [20, 700, 1320, 100], ...
                        'FontSize', 12, 'FontWeight', 'bold');

    start_btn = uibutton(control_panel, 'Text', 'Start Monitoring', ...
                       'Position', [20, 40, 120, 30], ...
                       'ButtonPushedFcn', @start_monitoring);

    stop_btn = uibutton(control_panel, 'Text', 'Stop Monitoring', ...
                      'Position', [160, 40, 120, 30], ...
                      'ButtonPushedFcn', @stop_monitoring);

    load_label = uilabel(control_panel, 'Text', 'Load Factor:', ...
                       'Position', [300, 45, 80, 20]);

    load_slider = uislider(control_panel, 'Position', [390, 50, 200, 3], ...
                        'Limits', [0.5, 1.5], 'Value', 1.0, ...
                        'ValueChangedFcn', @adjust_load);

    load_value = uilabel(control_panel, 'Text', '1.00', ...
                       'Position', [600, 45, 40, 20]);

    status_panel = uipanel(parent, 'Title', 'System Status', ...
                        'Position', [20, 550, 400, 130], ...
                        'FontSize', 12, 'FontWeight', 'bold');

    voltage_lamp = uilamp(status_panel, 'Position', [20, 80, 20, 20], ...
                        'Color', 'green');
    voltage_status = uilabel(status_panel, 'Text', 'Voltage: Normal', ...
                           'Position', [50, 80, 120, 20]);

    frequency_lamp = uilamp(status_panel, 'Position', [20, 50, 20, 20], ...
                          'Color', 'green');
    frequency_status = uilabel(status_panel, 'Text', 'Frequency: 60.0 Hz', ...
                             'Position', [50, 50, 120, 20]);

    loading_lamp = uilamp(status_panel, 'Position', [20, 20, 20, 20], ...
                        'Color', 'yellow');
    loading_status = uilabel(status_panel, 'Text', 'Loading: 82.8%', ...
                           'Position', [50, 20, 120, 20]);

    voltage_trend_panel = uipanel(parent, 'Title', 'Voltage Trend', ...
                                'Position', [440, 400, 440, 280], ...
                                'FontSize', 12, 'FontWeight', 'bold');

    ax_voltage = uiaxes(voltage_trend_panel, 'Position', [20, 20, 400, 240]);
    ax_voltage.Title.String = 'Bus Voltage Monitoring';
    ax_voltage.XLabel.String = 'Time (s)';
    ax_voltage.YLabel.String = 'Voltage (p.u.)';
```

```
    grid(ax_voltage, 'on');

    power_trend_panel = uipanel(parent, 'Title', 'Power Flow Trend', ...
                                'Position', [900, 400, 440, 280], ...
                                'FontSize', 12, 'FontWeight', 'bold');

    ax_power = uiaxes(power_trend_panel, 'Position', [20, 20, 400, 240]);
    ax_power.Title.String = 'Power Flow Monitoring';
    ax_power.XLabel.String = 'Time (s)';
    ax_power.YLabel.String = 'Power (MW)';
    grid(ax_power, 'on');

    alarm_panel = uipanel(parent, 'Title', 'Alarms & Events', ...
                          'Position', [20, 50, 1320, 330], ...
                          'FontSize', 12, 'FontWeight', 'bold');

    alarm_text = {
        '[06-07-2025 22:29:31] System initialized'
        '[06-07-2025 22:29:32] Load flow analysis completed - 4 iterations'
        '[06-07-2025 22:29:33] Warning: Under voltage at SUBSTATION BUS (94.6%)'
        '[06-07-2025 22:29:33] Warning: Under voltage at T1_LS (94.6%)'
        '[06-07-2025 22:29:34] Transformer T1 loading: 82.8% (Normal)'
        '[06-07-2025 22:29:35] Cable loading: 56.4% (Normal)'
        '[06-07-2025 22:29:36] System stable - All parameters within limits'
    };

    alarm_display = uitextarea(alarm_panel, 'Value', alarm_text, ...
                               'Position', [20, 20, 1280, 290], ...
                               'Editable', 'off');

    setappdata(parent, 'start_btn', start_btn);
    setappdata(parent, 'stop_btn', stop_btn);
    setappdata(parent, 'load_slider', load_slider);
    setappdata(parent, 'load_value', load_value);
    setappdata(parent, 'ax_voltage', ax_voltage);
    setappdata(parent, 'ax_power', ax_power);
    setappdata(parent, 'alarm_display', alarm_display);
    setappdata(parent, 'voltage_lamp', voltage_lamp);
    setappdata(parent, 'frequency_lamp', frequency_lamp);
    setappdata(parent, 'loading_lamp', loading_lamp);
    setappdata(parent, 'voltage_status', voltage_status);
    setappdata(parent, 'frequency_status', frequency_status);
    setappdata(parent, 'loading_status', loading_status);
end

function draw_single_line_diagram(ax)
    cla(ax);
    hold(ax, 'on');

    % Bus positions
    buses = struct();
```

```
    buses.T_sub = [1, 5];
    buses.T1_HS = [3, 5];
    buses.T1_LS = [5, 3];
    buses.SUBSTATION_BUS = [7, 3];

    % Buses
    bus_names = fieldnames(buses);
    for i = 1:length(bus_names)
        pos = buses.(bus_names{i});
        plot(ax, pos(1), pos(2), 'ks', 'MarkerSize', 8, 'MarkerFaceColor', 'black');
        text(ax, pos(1), pos(2)+0.2, bus_names{i}, 'HorizontalAlignment', 'center',↙
'FontSize', 8);
    end

    % Transmission line (T_sub to T1_HS)
    plot(ax, [buses.T_sub(1), buses.T1_HS(1)], [buses.T_sub(2), buses.T1_HS(2)], 'b-',↙
'LineWidth', 2);
    text(ax, 2, 5.2, 'Line1', 'HorizontalAlignment', 'center', 'FontSize', 8);

    % Transformer (T1_HS to T1_LS)
    plot(ax, [buses.T1_HS(1), buses.T1_LS(1)], [buses.T1_HS(2), buses.T1_LS(2)], 'r-',↙
'LineWidth', 2);
    plot(ax, 4, 4, 'ro', 'MarkerSize', 15, 'MarkerFaceColor', 'red');
    text(ax, 4, 4.3, 'T1', 'HorizontalAlignment', 'center', 'FontSize', 8, 'FontWeight',↙
'bold');
    text(ax, 4, 3.7, '100MVA', 'HorizontalAlignment', 'center', 'FontSize', 7);

    % Underground cable (T1_LS to SUBSTATION_BUS)
    plot(ax, [buses.T1_LS(1), buses.SUBSTATION_BUS(1)], [buses.T1_LS(2), buses.↙
SUBSTATION_BUS(2)], 'g-', 'LineWidth', 2);
    text(ax, 6, 3.2, 'UG CABLE', 'HorizontalAlignment', 'center', 'FontSize', 8);

    % Loads
    % Feeder 1
    plot(ax, [buses.SUBSTATION_BUS(1), buses.SUBSTATION_BUS(1)+0.5], [buses.↙
SUBSTATION_BUS(2), buses.SUBSTATION_BUS(2)-0.5], 'k-', 'LineWidth', 1);
    plot(ax, buses.SUBSTATION_BUS(1)+0.5, buses.SUBSTATION_BUS(2)-0.5, 'mo',↙
'MarkerSize', 10, 'MarkerFaceColor', 'magenta');
    text(ax, buses.SUBSTATION_BUS(1)+0.5, buses.SUBSTATION_BUS(2)-0.8, 'Feeder1',↙
'HorizontalAlignment', 'center', 'FontSize', 7);
    text(ax, buses.SUBSTATION_BUS(1)+0.5, buses.SUBSTATION_BUS(2)-1.0, '40MVA',↙
'HorizontalAlignment', 'center', 'FontSize', 7);

    % Feeder 2
    plot(ax, [buses.SUBSTATION_BUS(1), buses.SUBSTATION_BUS(1)+0.5], [buses.↙
SUBSTATION_BUS(2), buses.SUBSTATION_BUS(2)+0.5], 'k-', 'LineWidth', 1);
    plot(ax, buses.SUBSTATION_BUS(1)+0.5, buses.SUBSTATION_BUS(2)+0.5, 'mo',↙
'MarkerSize', 10, 'MarkerFaceColor', 'magenta');
    text(ax, buses.SUBSTATION_BUS(1)+0.5, buses.SUBSTATION_BUS(2)+0.8, 'Feeder2',↙
'HorizontalAlignment', 'center', 'FontSize', 7);
    text(ax, buses.SUBSTATION_BUS(1)+0.5, buses.SUBSTATION_BUS(2)+1.0, '40MVA',↙
```

```
'HorizontalAlignment', 'center', 'FontSize', 7);

    % Utility source
    plot(ax, buses.T_sub(1)-0.5, buses.T_sub(2), 'ko', 'MarkerSize', 15,↙
'MarkerFaceColor', 'cyan');
    text(ax, buses.T_sub(1)-0.5, buses.T_sub(2)+0.3, 'Utility', 'HorizontalAlignment',↙
'center', 'FontSize', 8, 'FontWeight', 'bold');
    text(ax, buses.T_sub(1)-0.5, buses.T_sub(2)-0.3, '230kV', 'HorizontalAlignment',↙
'center', 'FontSize', 7);

    % Voltage levels
    text(ax, buses.T_sub(1), buses.T_sub(2)-0.3, '230kV', 'HorizontalAlignment',↙
'center', 'FontSize', 7, 'Color', 'blue');
    text(ax, buses.T1_HS(1), buses.T1_HS(2)-0.3, '230kV', 'HorizontalAlignment',↙
'center', 'FontSize', 7, 'Color', 'blue');
    text(ax, buses.T1_LS(1), buses.T1_LS(2)-0.3, '13.8kV', 'HorizontalAlignment',↙
'center', 'FontSize', 7, 'Color', 'blue');
    text(ax, buses.SUBSTATION_BUS(1), buses.SUBSTATION_BUS(2)-0.3, '13.8kV',↙
'HorizontalAlignment', 'center', 'FontSize', 7, 'Color', 'blue');


    ax.XLim = [0, 8.5];
    ax.YLim = [1.5, 6];
    ax.XTick = [];
    ax.YTick = [];
    ax.Title.String = 'Microgrid Single Line Diagram';
    grid(ax, 'off');
    axis(ax, 'equal');
end

function start_monitoring(src, event)
    parent = src.Parent.Parent;
    ax_voltage = getappdata(parent, 'ax_voltage');
    ax_power = getappdata(parent, 'ax_power');

    t = 0:0.1:30;

    voltage_base = 0.946;
    voltage_variation = voltage_base + 0.02*sin(2*pi*0.1*t) + 0.01*randn(size(t));

    power_base = 62.64;
    power_variation = power_base + 5*sin(2*pi*0.05*t) + 2*randn(size(t));

    plot(ax_voltage, t, voltage_variation, 'b-', 'LineWidth', 1.5);
    ax_voltage.YLim = [0.9, 1.0];

    plot(ax_power, t, power_variation, 'r-', 'LineWidth', 1.5);
    ax_power.YLim = [50, 75];

    voltage_lamp = getappdata(parent, 'voltage_lamp');
    if min(voltage_variation) < 0.95
```

```
        voltage_lamp.Color = 'red';
    else
        voltage_lamp.Color = 'green';
    end
end

function stop_monitoring(src, event)
    parent = src.Parent.Parent;
    ax_voltage = getappdata(parent, 'ax_voltage');
    ax_power = getappdata(parent, 'ax_power');

    cla(ax_voltage);
    cla(ax_power);

    ax_voltage.Title.String = 'Bus Voltage Monitoring - Stopped';
    ax_power.Title.String = 'Power Flow Monitoring - Stopped';
end

function adjust_load(src, event)
    parent = src.Parent.Parent;
    load_value = getappdata(parent, 'load_value');
    loading_status = getappdata(parent, 'loading_status');
    loading_lamp = getappdata(parent, 'loading_lamp');

    load_factor = src.Value;
    load_value.Text = sprintf('%.2f', load_factor);

    base_loading = 82.8;
    new_loading = base_loading * load_factor;
    loading_status.Text = sprintf('Loading: %.1f%%', new_loading);

    if new_loading > 95
        loading_lamp.Color = 'red';
    elseif new_loading > 85
        loading_lamp.Color = 'yellow';
    else
        loading_lamp.Color = 'green';
    end
end

microgrid_dashboard();
```