```
classdef signalPlotter < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                matlab.ui.Figure
        RESETButton             matlab.ui.control.Button
        TextArea_2              matlab.ui.control.TextArea
        EquationoftheplotLabel  matlab.ui.control.Label
        SelectasignaltypeLabel  matlab.ui.control.Label
        SelectawaveformLabel    matlab.ui.control.Label
        TextArea                matlab.ui.control.TextArea
        TypeDropDown            matlab.ui.control.DropDown
        TypeDropDownLabel       matlab.ui.control.Label
        WaveformDropDown        matlab.ui.control.DropDown
        WaveformDropDownLabel   matlab.ui.control.Label
        SignalPlotterLabel      matlab.ui.control.Label
        PLOTButton              matlab.ui.control.Button
        UIAxes                  matlab.ui.control.UIAxes
    end

    properties (Access = public)
        value1          % type of waveform
        value2          % type of signal (CTS / DTS)
        y               % equation for plotting
        x               % misc
        t1              % misc
        t2              % misc
        sliderVal       % store value of slider for exponential signal
        p               % misc
        equation        % display equation
        x1              % misc
    end

    % Callbacks that handle component events
    methods (Access = private)

        % Callback function
        function SliderValueChanged(app, event)

            value = app.Slider.Value;
            app.sliderVal = value;

        end

        % Value changed function: WaveformDropDown
        function WaveformDropDownValueChanged(app, event)

            value = app.WaveformDropDown.Value;
            app.value1 = value;

            app.t1 = 0 : 1;
            app.t2 = -1 : 0.01 : 1;
            app.x = 0 : 0.1: 2 * pi;
            app.x1 = linspace(-5, 5);
            app.p = -2 * pi : 0.1 : 2 * pi;


            if app.value1 == "Step"
                app.y = app.t2 >= 0;
            elseif app.value1 == "Ramp"
                app.y = app.x;
            elseif app.value1 == "Parabolic"
                app.y = sqrt(app.x);
            elseif app.value1 == "Sinc"
                app.y = sinc(app.x1);
            elseif app.value1 == "Impulse"
                app.y = app.t1 == 0;
            elseif app.value1 == "Sine"
                app.y = sin(app.x);
            elseif app.value1 == "Exponential"
                app.y = exp(app.t2);
            elseif app.value1 == "Signum"
                app.y = sign((-1 : 0.1 : 1));
            end

        end

        % Value changed function: TypeDropDown
        function TypeDropDownValueChanged(app, event)

            value = app.TypeDropDown.Value;
            app.value2 = value;

        end

        % Value changed function: TextArea
        function TextAreaValueChanged(app, event)
            value = app.TextArea.Value;
            app.equation = value;
        end

        % Button pushed function: PLOTButton
        function PLOTButtonPushed(app, event)

            if strcmp(app.value2, "CTS")
                if strcmp(app.value1, "Impulse")
                    stem(app.UIAxes, app.y)
                else
                    if strcmp(app.value1, "Sinc")
                        plot(app.UIAxes, app.x1, app.y);
                    elseif strcmp(app.value1, "Signum")
                        plot(app.UIAxes, [-1 : 0.1 : 1], app.y);
```

```
                    else
                        plot(app.UIAxes, app.y);
                    end
                end
            elseif strcmp(app.value2, "DTS")
                if strcmp(app.value1, "Sinc")
                    stem(app.UIAxes, app.x1, app.y);
                elseif strcmp(app.value1, "Signum")
                    stem(app.UIAxes, [-1 : 0.1 : 1], app.y);
                else
                    stem(app.UIAxes, app.y);
                end
            end

            if strcmp(app.value2, "CTS")
                if app.value1 == "Step"
                    app.TextArea.Value = sprintf("y = f(t) = [t] for t ∈ R");
                elseif app.value1 == "Ramp"
                    app.TextArea.Value = sprintf("r(t) = kt, t = 0; " + ...
                        "r(t) = 0, t != 0");
                elseif app.value1 == "Parabolic"
                    app.TextArea.Value = sprintf("y^2 = x");
                elseif app.value1 == "Sine"
                    app.TextArea.Value = sprintf("y = sin(wt)");
                elseif app.value1 == "Impulse"
                    app.TextArea.Value = sprintf("δ(t) = 1, t = 0; " + ...
                        "δ(t) = 0, t != 0");
                elseif app.value1 == "Exponential"
                    app.TextArea.Value = sprintf("y = e^t");
                elseif app.value1 == "Signum"
                    app.TextArea.Value = sprintf("x(t) = 1, t >= 0; " + ...
                        "x(t) = -1, t < 0");
                elseif app.value1 == "Sinc"
                    app.TextArea.Value = sprintf("y = sinc = sin(pi * t) / (pi * t)");
                end
            elseif strcmp(app.value2, "DTS")
                if app.value1 == "Step"
                    app.TextArea.Value = sprintf("y = f(n) = [n] for n ∈ R");
                elseif app.value1 == "Ramp"
                    app.TextArea.Value = sprintf("r(n) = kn, n = 0; " + ...
                        "r(n) = 0, n != 0");
                elseif app.value1 == "Parabolic"
                    app.TextArea.Value = sprintf("y^2 = x");
                elseif app.value1 == "Sine"
                    app.TextArea.Value = sprintf("y = sin(wn)");
                elseif app.value1 == "Impulse"
                    app.TextArea.Value = sprintf("δ(n) = 1, n = 0; " + ...
                        "δ(n) = 0, n != 0");
                elseif app.value1 == "Exponential"
                    app.TextArea.Value = sprintf("y = e^n");
                elseif app.value1 == "Signum"
                    app.TextArea.Value = sprintf("x(n) = 1, n >= 0; " + ...
                        "x(n) = -1, n < 0");
                elseif app.value1 == "Sinc"
                    app.TextArea.Value = sprintf("y = sinc = sin(pi * n) / (pi * n)");
                end

            end

        end

        % Button pushed function: RESETButton
        function RESETButtonPushed(app, event)
            cla(app.UIAxes);
            app.TextArea.Value = sprintf("");
        end
    end

    % Component initialization
    methods (Access = private)

        % Create UIFigure and components
        function createComponents(app)

            % Create UIFigure and hide until all components are created
            app.UIFigure = uifigure('Visible', 'off');
            app.UIFigure.Color = [0.8 0.8 0.8];
            app.UIFigure.Position = [100 100 652 502];
            app.UIFigure.Name = 'MATLAB App';

            % Create UIAxes
            app.UIAxes = uiaxes(app.UIFigure);
            title(app.UIAxes, 'Signal')
            xlabel(app.UIAxes, 'Time')
            ylabel(app.UIAxes, 'Amplitude ')
            zlabel(app.UIAxes, 'Z')
            app.UIAxes.FontWeight = 'bold';
            app.UIAxes.XGrid = 'on';
            app.UIAxes.YGrid = 'on';
            app.UIAxes.Position = [243 45 390 261];

            % Create PLOTButton
            app.PLOTButton = uibutton(app.UIFigure, 'push');
            app.PLOTButton.ButtonPushedFcn = createCallbackFcn(app, @PLOTButtonPushed, true);
            app.PLOTButton.FontWeight = 'bold';
            app.PLOTButton.Position = [280 317 127 32];
            app.PLOTButton.Text = 'PLOT';

            % Create SignalPlotterLabel
            app.SignalPlotterLabel = uilabel(app.UIFigure);
            app.SignalPlotterLabel.HorizontalAlignment = 'center';
            app.SignalPlotterLabel.FontSize = 24;
            app.SignalPlotterLabel.FontWeight = 'bold';
            app.SignalPlotterLabel.Position = [180 448 328 31];
```

```
            app.SignalPlotterLabel.Text = 'Signal Plotter';

            % Create WaveformDropDownLabel
            app.WaveformDropDownLabel = uilabel(app.UIFigure);
            app.WaveformDropDownLabel.HorizontalAlignment = 'right';
            app.WaveformDropDownLabel.FontSize = 15;
            app.WaveformDropDownLabel.FontWeight = 'bold';
            app.WaveformDropDownLabel.Position = [45 372 77 22];
            app.WaveformDropDownLabel.Text = 'Waveform';

            % Create WaveformDropDown
            app.WaveformDropDown = uidropdown(app.UIFigure);
            app.WaveformDropDown.Items = {'Step', 'Ramp', 'Parabolic', 'Sine', 'Impulse', 'Exponential', 'Signum', 'Sinc'};
            app.WaveformDropDown.ValueChangedFcn = createCallbackFcn(app, @WaveformDropDownValueChanged, true);
            app.WaveformDropDown.FontSize = 15;
            app.WaveformDropDown.Position = [137 367 144 27];
            app.WaveformDropDown.Value = 'Step';

            % Create TypeDropDownLabel
            app.TypeDropDownLabel = uilabel(app.UIFigure);
            app.TypeDropDownLabel.HorizontalAlignment = 'right';
            app.TypeDropDownLabel.FontSize = 15;
            app.TypeDropDownLabel.FontWeight = 'bold';
            app.TypeDropDownLabel.Position = [397 374 39 22];
            app.TypeDropDownLabel.Text = 'Type';

            % Create TypeDropDown
            app.TypeDropDown = uidropdown(app.UIFigure);
            app.TypeDropDown.Items = {'CTS', 'DTS'};
            app.TypeDropDown.ValueChangedFcn = createCallbackFcn(app, @TypeDropDownValueChanged, true);
            app.TypeDropDown.FontSize = 15;
            app.TypeDropDown.Position = [451 374 152 22];
            app.TypeDropDown.Value = 'CTS';

            % Create TextArea
            app.TextArea = uitextarea(app.UIFigure);
            app.TextArea.ValueChangedFcn = createCallbackFcn(app, @TextAreaValueChanged, true);
            app.TextArea.Editable = 'off';
            app.TextArea.HorizontalAlignment = 'center';
            app.TextArea.FontSize = 15;
            app.TextArea.Position = [21 169 214 48];

            % Create SelectawaveformLabel
            app.SelectawaveformLabel = uilabel(app.UIFigure);
            app.SelectawaveformLabel.FontSize = 13;
            app.SelectawaveformLabel.FontWeight = 'bold';
            app.SelectawaveformLabel.FontAngle = 'italic';
            app.SelectawaveformLabel.Position = [109 410 118 22];
            app.SelectawaveformLabel.Text = 'Select a waveform';

            % Create SelectasignaltypeLabel
            app.SelectasignaltypeLabel = uilabel(app.UIFigure);
            app.SelectasignaltypeLabel.FontSize = 13;
            app.SelectasignaltypeLabel.FontWeight = 'bold';
            app.SelectasignaltypeLabel.FontAngle = 'italic';
            app.SelectasignaltypeLabel.Position = [437 410 126 22];
            app.SelectasignaltypeLabel.Text = 'Select a signal type';

            % Create EquationoftheplotLabel
            app.EquationoftheplotLabel = uilabel(app.UIFigure);
            app.EquationoftheplotLabel.FontSize = 15;
            app.EquationoftheplotLabel.FontWeight = 'bold';
            app.EquationoftheplotLabel.Position = [55 233 146 22];
            app.EquationoftheplotLabel.Text = 'Equation of the plot';

            % Create TextArea_2
            app.TextArea_2 = uitextarea(app.UIFigure);
            app.TextArea_2.Editable = 'off';
            app.TextArea_2.HorizontalAlignment = 'center';
            app.TextArea_2.FontSize = 10;
            app.TextArea_2.Position = [29 29 198 101];
            app.TextArea_2.Value = {'GROUP 1'; 'Aaryan Shakti - 3001'; 'Abhishek Singh - 3002'; 'Abhishek Yadav - 3003'; 'Aditi Sathe - 3004'};

            % Create RESETButton
            app.RESETButton = uibutton(app.UIFigure, 'push');
            app.RESETButton.ButtonPushedFcn = createCallbackFcn(app, @RESETButtonPushed, true);
            app.RESETButton.FontWeight = 'bold';
            app.RESETButton.Position = [397 10 118 28];
            app.RESETButton.Text = 'RESET';

            % Show the figure after all components are created
            app.UIFigure.Visible = 'on';
        end
    end

    % App creation and deletion
    methods (Access = public)

        % Construct app
        function app = signalPlotter

            % Create UIFigure and components
            createComponents(app)

            % Register the app with App Designer
            registerApp(app, app.UIFigure)

            if nargout == 0
                clear app
            end
        end

        % Code that executes before app deletion
        function delete(app)
```

```
            % Delete UIFigure when app is deleted
            delete(app.UIFigure)
        end
    end
end
```