function The\_Solar\_Solution\_GUI()

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%% Script for Solar Solution GUI %%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Initial Clearing (TAB JUMP)

% Clear and close all operations and variables

clear all

close all

clc

% This function docks the figures...

set(0,'DefaultFigureWindowStyle','normal')

% reverse by using "normal"

set(0,'DefaultFigureVisible','on');

%% Load Data

% NASA Data

solar\_psh\_data = importdata('PSH\_NASA\_Data.mat');

% AUS Government Data

kwhr\_avg\_data = importdata('Daily\_Usage\_Gov\_Data.mat');

%% Find Screen Size and Calculate Window

% Set Number of tabs and tab labels

NumTabs = 8; % Number of tabs to be generated

TabLabels = {'Data Aquisition'; 'Input Data'; 'Estimated Production'; 'Finance Options'; 'Display';'2x Production Graph';'2x Yearly Graph';'5x Pie Graph';};

if size(TabLabels,1) ~= NumTabs

errordlg('Number of tabs and tab labels must be the same','Setup Error');

return

end

% Get user screen size

SC = get(0, 'ScreenSize');

MaxMonitorX = SC(3);

MaxMonitorY = SC(4);

% Set the figure window size values

MainFigScale = .8; % Change this value to adjust the figure size

MaxWindowX = round(MaxMonitorX\*MainFigScale);

MaxWindowY = round(MaxMonitorY\*MainFigScale);

XBorder = (MaxMonitorX-MaxWindowX)/2;

YBorder = (MaxMonitorY-MaxWindowY)/2;

TabOffset = 0; % This value offsets the tabs inside the figure.

ButtonHeight = 40;

PanelWidth = MaxWindowX-2\*TabOffset+4;

PanelHeight = MaxWindowY-ButtonHeight-2\*TabOffset;

ButtonWidth = round((PanelWidth-NumTabs)/NumTabs);

% Set the color varables

White = [1 1 1]; % White - Selected tab color

Grey = .9\*White; % Light Grey - Background color

%% Create a Figure for GUI

% Creates the maine figure for the GUI which all handles use

hTabFig = figure(...

'Units', 'pixels',...

'Toolbar', 'none',...

'Position',[ XBorder, YBorder, MaxWindowX, MaxWindowY ],...

'NumberTitle', 'off',...

'Name', 'Solar Solution',...

'MenuBar', 'none',...

'Resize', 'off',...

'DockControls', 'off',...

'Color', White);

%% Define a cell array for panel and pushbutton handles, pushbuttons labels and other data

% Rows are for each tab + two additional rows for other data

% Columns are uipanel handles, selection pushbutton handles, and tab label strings - 3 columns.

TabHandles = cell(NumTabs,3);

TabHandles(:,3) = TabLabels(:,1);

% Add additional rows for other data

TabHandles{NumTabs+1,1} = hTabFig; % Main figure handle

TabHandles{NumTabs+1,2} = PanelWidth; % Width of tab panel

TabHandles{NumTabs+1,3} = PanelHeight; % Height of tab panel

TabHandles{NumTabs+2,1} = 0; % Handle to default tab 2 content(set later)

TabHandles{NumTabs+2,2} = White; % Selected tab Color

TabHandles{NumTabs+2,3} = Grey; % Background color

%% Build the Tabs

%Creates tabs for the GUI in a for-loop

for TabNumber = 1:NumTabs

% Create a UIPanel

TabHandles{TabNumber,1} = uipanel('Units', 'pixels', ...

'Visible', 'off', ...

'Backgroundcolor', White, ...

'BorderWidth',1, ...

'Position', [TabOffset TabOffset ...

PanelWidth PanelHeight]);

% Create a selection pushbutton

TabHandles{TabNumber,2} = uicontrol('Style', 'pushbutton',...

'Units', 'pixels', ...

'BackgroundColor', Grey, ...

'Position', [TabOffset+(TabNumber-1)\*ButtonWidth PanelHeight+TabOffset...

ButtonWidth ButtonHeight], ...

'String', TabHandles{TabNumber,3},...

'HorizontalAlignment', 'center',...

'FontName', 'arial',...

'FontWeight', 'bold',...

'FontSize', 10);

end

%% Define the callbacks for the Tab Buttons, Clock, Prefill and Reset Button

% All callbacks go to the same function with the additional argument being the Tab number

for CountTabs = 1:NumTabs

set(TabHandles{CountTabs,2}, 'callback', ...

{@TabSellectCallback, CountTabs});

end

% Set up the time

clock\_disp();

% Create a reset button

for count = 1:1:NumTabs

reset\_button = uicontrol('Units', 'normalized', 'Position',[0.95 0 0.05 0.05], 'Style', 'pushbutton',...

'String', 'Reset', 'Visible', 'On','Callback', @reset,'Parent', TabHandles{count,1},...

'Backgroundcolor',Grey, 'Foregroundcolor', 'black', 'FontSize', 15);

end

%{

NOTE: This button is used for testing the program and fault finding slash error checking

Create a Prefill button

%}

prefill\_button = uicontrol('Units', 'normalized', 'Position',[0.95 0.05 0.05 0.05], 'Style', 'pushbutton',...

'String', 'E.X', 'Visible', 'On','Callback', @prefill,'Parent', TabHandles{1,1},...

'Backgroundcolor',Grey, 'Foregroundcolor', 'black', 'FontSize', 15);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Tab 1 Content: PROMPTING TAB %%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Create tab reference for parent handles

prompt\_page = 1;

persistent index;

% Define Max number of inputs for progress bar

persistent number\_of\_inputs;

number\_of\_inputs = 13;

%% Entry Prompts

% Create standard size for questions buttons and positions

standard\_question = [0.5-0.35/2 0.7 0.35 0.12];

standard\_yes = [0.2 0.4 0.2 0.2];

standard\_no = [0.6 0.4 0.2 0.2];

% Create a button to enter calculator

enter\_gui\_button = uicontrol('Units', 'normalized', 'Position',[0.375 0.4 0.25 0.25], 'Style', 'pushbutton',...

'String', 'Enter Solar Solution', 'Visible', 'On','Callback', @entry\_click,'Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor',Grey, 'Foregroundcolor', 'black', 'FontSize', 20);

% Create function for entry

function entry\_click(~, eventdata)

set(enter\_gui\_button,'Visible','OFF')

set(gas\_mains\_question,'Visible','ON')

set(button\_yes\_gas\_mains,'Visible','ON')

set(button\_no\_gas\_mains,'Visible','ON')

set(prefill\_button,'Visible','On')

progress\_bar(1);

end

%% Gas Connected

% Create the gas related questions

gas\_mains\_question = uicontrol('Units', 'normalized', 'Position',standard\_question, 'Style', 'text',...

'String', 'Do you have gas connected mains?','Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20, 'Visible', 'OFF');

% Create button if has gas

button\_yes\_gas\_mains = uicontrol('Units', 'normalized', 'Position',standard\_yes, 'Style', 'pushbutton',...

'String', 'Yes', 'Visible', 'On','Callback', @gas\_click,'Parent', TabHandles{prompt\_page,1},'tag', 'gas\_yes',...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20, 'Visible', 'OFF');

% Create button if has no gas

button\_no\_gas\_mains = uicontrol('Units', 'normalized', 'Position',standard\_no, 'Style', 'pushbutton',...

'String', 'No', 'Visible', 'On','Callback', @gas\_click,'Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20, 'Visible', 'OFF');

% Create persistent variables

persistent gas\_mains\_input;

% Create function for entry

function gas\_click(hObject, eventdata)

% Set previous questions off and next on with progress

set(gas\_mains\_question,'Visible','Off')

set(button\_yes\_gas\_mains,'Visible','Off')

set(button\_no\_gas\_mains,'Visible','Off')

% Find the answer and set output

string = get(hObject, 'tag');

if strcmp(string, 'gas\_yes') == 1

gas\_mains\_input = 1

set(gas\_main\_value, 'String', 'Yes')

else

set(gas\_main\_value, 'String', 'No')

gas\_mains\_input = 0

end

% Set next questions on

set(pool\_question,'Visible','ON')

set(button\_yes\_pool,'Visible','ON')

set(button\_no\_pool,'Visible','ON')

progress\_bar(2);

end

%% Pool Connected

% Create the pool related questions

pool\_question = uicontrol('Units', 'normalized', 'Position',standard\_question, 'Style', 'text',...

'String', 'Do you have a pool?','Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20, 'Visible', 'OFF');

% Create if yes for pool

button\_yes\_pool = uicontrol('Units', 'normalized', 'Position',standard\_yes, 'Style', 'pushbutton',...

'String', 'Yes', 'Visible', 'On','Callback', @pool\_click,'Parent', TabHandles{prompt\_page,1},'tag', 'pool\_yes',...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20, 'Visible', 'OFF');

% Create if no for pool

button\_no\_pool = uicontrol('Units', 'normalized', 'Position',standard\_no, 'Style', 'pushbutton',...

'String', 'No', 'Visible', 'On','Callback', @pool\_click,'Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20, 'Visible', 'OFF');

% Create persistent variables

persistent pool\_input;

% Create function for entry

function pool\_click(hObject, eventdata)

% Set previous questions off

set(pool\_question,'Visible','Off')

set(button\_yes\_pool,'Visible','Off')

set(button\_no\_pool,'Visible','Off')

% Find the answer and set output

string = get(hObject, 'tag');

if strcmp(string, 'pool\_yes') == 1

pool\_input = 1

set(pool\_value, 'String', 'Yes')

else

set(pool\_value, 'String', 'NO')

pool\_input = 0

end

set(solar\_size\_value, 'String', '0')

set(solar\_cost\_value, 'String', '0')

set(battery\_size\_value, 'String', '0')

set(battery\_cost\_value, 'String', '0')

% Set next questions on

set(percentage\_question,'Visible','ON')

set(percentage\_popupmenu,'Visible','ON')

progress\_bar(3);

end

%% Percentage Usage

% Create the related questions

percentage\_question = uicontrol('Units', 'normalized', 'Position',standard\_question, 'Style', 'text',...

'String', 'What percentage of your electricity do you use during the day?','Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20, 'Visible', 'Off');

% Create persistent variables

persistent percentage\_input

percentage\_input = 0;

% Create function selection click

function percentage\_next(hObject, eventdata)

% Create a next button once an option has been selected

set(percentage\_next\_button,'Visible','On')

% Find the answer and set output

index = get(hObject, 'Value');

percentage\_input = percentage\_list(index)

end

% Dropdown list

percentage\_list = [20 25 30 35 40 45];

% Set up pop up menu with pulldown data

percentage\_popupmenu = uicontrol('Units', 'normalized', 'Position', [0.35 0.5 0.3 0.15], 'Style', 'popupmenu','Parent', TabHandles{prompt\_page,1},...

'String', percentage\_list,'Callback', @percentage\_next, 'tag', 'KW\_menu', 'Visible', 'OFF', 'FontSize', 20);

% Create button for next click

percentage\_next\_button = uicontrol('Units', 'normalized', 'Position',[0.35 0.4 0.3 0.1], 'Style', 'pushbutton',...

'String', 'Next', 'Visible', 'On','Callback', @percentage\_next\_click, 'Visible', 'On','Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20, 'Visible', 'OFF');

% Create function for click

function percentage\_next\_click(hObject, eventdata)

% Set previous questions off and next on with progress

set(percentage\_question,'Visible','Off')

set(percentage\_popupmenu,'Visible','Off')

set(percentage\_next\_button,'Visible','Off')

set(performance\_question,'Visible','On')

set(performance\_popupmenu,'Visible','On')

progress\_bar(4) ;

end

%% Performance Ratio

% Create the related questions

performance\_question = uicontrol('Units', 'normalized', 'Position',standard\_question, 'Style', 'text',...

'String', 'What is your performance ratio of the solar system?','Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20, 'Visible', 'OFF');

% Create persistent variables

persistent performance\_input

performance\_input = 0;

% Create function selection click

function performance\_next(hObject, eventdata)

% Create a next button once an option has been selected

set(performance\_next\_button,'Visible','On')

% Find the answer and set output

index = get(hObject, 'Value');

performance\_input = performance\_list(index)

end

% Dropdown list

performance\_list = [0.70 0.75 0.80 0.85 0.90];

% Set up pop up menu with pulldown data

performance\_popupmenu = uicontrol('Units', 'normalized', 'Position', [0.35 0.5 0.3 0.15], 'Style', 'popupmenu','Parent', TabHandles{prompt\_page,1},...

'String', performance\_list,'Callback', @performance\_next, 'tag', 'KW\_menu', 'Visible', 'OFF', 'FontSize', 20);

% Create button for next click

performance\_next\_button = uicontrol('Units', 'normalized', 'Position',[0.35 0.4 0.3 0.1], 'Style', 'pushbutton',...

'String', 'Next', 'Visible', 'On','Callback', @performance\_next\_click, 'Visible', 'On','Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20, 'Visible', 'OFF');

% Create function for click

function performance\_next\_click(hObject, eventdata)

% Set previous questions off and next on with progress

set(performance\_question,'Visible','Off')

set(performance\_popupmenu,'Visible','Off')

set(performance\_next\_button,'Visible','Off')

set(text\_solar\_question,'Visible','ON')

set(button\_yes\_solar,'Visible','ON')

set(button\_no\_solar,'Visible','ON')

progress\_bar(5);

end

%% Solar Installed

% Create the related questions

text\_solar\_question = uicontrol('Units', 'normalized', 'Position',standard\_question, 'Style', 'text',...

'String', 'Do you have a solar system?', 'Visible', 'On','Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20, 'Visible', 'OFF');

% Create button if yes

button\_yes\_solar = uicontrol('Units', 'normalized', 'Position',standard\_yes, 'Style', 'pushbutton',...

'String', 'Yes', 'Visible', 'On','Callback', @solar\_click\_yes, 'Visible', 'On','Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20, 'Visible', 'OFF');

% Create button if no

button\_no\_solar = uicontrol('Units', 'normalized', 'Position',standard\_no, 'Style', 'pushbutton',...

'String', 'No', 'Visible', 'On','Callback', @solar\_click\_no, 'Visible', 'On','Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20, 'Visible', 'OFF');

% Create the size question

solar\_size\_question = uicontrol('Units', 'normalized', 'Position',standard\_question, 'Style', 'text',...

'String', 'What is the size of your solar system (kW)?', 'Visible', 'On','Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20, 'Visible', 'OFF');

% Create persistent variables

persistent solar\_installed

persistent solar\_size\_input

solar\_size\_input = 0;

% Create function selection click

function solar\_click\_yes(hObject, eventdata)

% Set previous questions off and next on with progress

set(text\_solar\_question,'Visible','OFF')

set(button\_yes\_solar,'Visible','OFF')

set(button\_no\_solar,'Visible','OFF')

% Find the answer and set output

solar\_installed = 1

set(solar\_size\_question,'Visible','ON')

set(KW\_popupmenu,'Visible','ON')

end

% Create function for click

function size\_next\_button(hObject, eventdata)

% Create a next button once an option has been selected

set(solar\_size\_next\_button,'Visible','ON')

% Find the answer and set output

index = get(hObject, 'Value');

solar\_size\_input = KW\_solar\_size(index)

set(solar\_size\_value, 'String', num2str(KW\_solar\_size(index)))

progress\_bar(5.25) ;

end

% Dropdown list

KW\_solar\_size = [1 3.5 5 7 9 15];

% Set up pop up menu with pulldown data

KW\_popupmenu = uicontrol('Units', 'normalized', 'Position', [0.35 0.5 0.3 0.15], 'Style', 'popupmenu','Parent', TabHandles{prompt\_page,1},...

'String', KW\_solar\_size,'Callback', @size\_next\_button, 'tag', 'KW\_menu', 'Visible', 'OFF', 'FontSize', 20);

% Create button for next click

solar\_size\_next\_button = uicontrol('Units', 'normalized', 'Position',[0.35 0.4 0.3 0.1], 'Style', 'pushbutton',...

'String', 'Next', 'Visible', 'On','Callback', @cost\_display, 'Visible', 'On','Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20, 'Visible', 'OFF');

% Create function for click

function cost\_display(hObject, eventdata)

% Set previous questions off and next on with progress

set(solar\_size\_next\_button,'Visible','off')

set(KW\_popupmenu,'Visible','off')

set(solar\_size\_question,'Visible','off')

set(cost\_solar\_question,'Visible','ON')

set(cost\_popupmenu,'Visible','ON')

end

%% Cost of Solar

% Create the related questions

cost\_solar\_question = uicontrol('Units', 'normalized', 'Position',standard\_question, 'Style', 'text',...

'String', 'How much did your solar system cost ($)?', 'Visible', 'On','Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20, 'Visible', 'OFF');

% Create persistent variables

persistent cost\_solar\_input

cost\_solar\_input = 0;

% Create function selection click

function cost\_next\_button\_call(hObject, eventdata)

% Set previous questions off and next on with progress

set(cost\_next\_button,'Visible','ON')

% Find the answer and set output

index = get(hObject, 'Value');

cost\_solar\_input = solar\_cost(index)

set(solar\_cost\_value, 'String', num2str(solar\_cost(index)))

progress\_bar(5.75);

end

% Dropdown list

solar\_cost = [4000 5000 6000 7000 8000 9000 10000 11000 13000 14000 16000];

% Set up pop up menu with pulldown data

cost\_popupmenu = uicontrol('Units', 'normalized', 'Position', [0.35 0.5 0.3 0.15], 'Style', 'popupmenu','Parent', TabHandles{prompt\_page,1},...

'String', solar\_cost,'Callback', @cost\_next\_button\_call, 'tag', 'cost\_menu', 'Visible', 'OFF', 'FontSize', 20);

% Create button if has exisiting solar

cost\_next\_button = uicontrol('Units', 'normalized', 'Position',[0.35 0.4 0.3 0.1], 'Style', 'pushbutton',...

'String', 'Next', 'Visible', 'Off','Callback', @solar\_click\_no,'Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20);

%% Battery Installed

% Create the related questions

text\_battery\_question = uicontrol('Units', 'normalized', 'Position',standard\_question, 'Style', 'text',...

'String', 'Do you have a battery energy storage (BES) system?', 'Visible', 'On','Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20, 'Visible', 'OFF');

% Create button if yes battery

button\_yes\_battery = uicontrol('Units', 'normalized', 'Position',standard\_yes, 'Style', 'pushbutton',...

'String', 'Yes', 'Visible', 'On','Callback', @battery\_click\_yes,'Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20, 'Visible', 'OFF');

% Create button if no battery

button\_no\_battery = uicontrol('Units', 'normalized', 'Position',standard\_no, 'Style', 'pushbutton',...

'String', 'No', 'Visible', 'On','Callback', @battery\_click\_no,'Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20, 'Visible', 'OFF');

% Create what size battery

battery\_size\_question = uicontrol('Units', 'normalized', 'Position',standard\_question, 'Style', 'text',...

'String', 'What is the size of your BES system (kWhr)?', 'Visible', 'On','Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20, 'Visible', 'OFF');

% Create persistent variables

persistent battery\_installed

battery\_installed = 0;

% Create function selection click

function solar\_click\_no(hObject, eventdata)

% Set previous questions off and next on with progress

set(text\_solar\_question,'Visible','OFF')

set(button\_yes\_solar,'Visible','OFF')

set(button\_no\_solar,'Visible','OFF')

set(cost\_solar\_question,'Visible','OFF')

set(cost\_popupmenu,'Visible','OFF')

set(cost\_next\_button,'Visible','OFF')

set(text\_battery\_question,'Visible','ON')

set(button\_yes\_battery,'Visible','ON')

set(button\_no\_battery,'Visible','ON')

progress\_bar(6);

end

% Create function for click

function battery\_click\_yes(hObject, eventdata)

set(text\_battery\_question,'Visible','OFF')

set(button\_yes\_battery,'Visible','OFF')

set(button\_no\_battery,'Visible','OFF')

% Find the answer and set output

battery\_installed = 1;

set(battery\_size\_question,'Visible','ON')

set(KWHR\_popupmenu,'Visible','ON')

end

% Dropdown list

KWHR\_battery\_size = [1 2 4 6 8 12];

% Set up pop up menu with pulldown data

KWHR\_popupmenu = uicontrol('Units', 'normalized', 'Position', [0.35 0.5 0.3 0.15], 'Style', 'popupmenu','Parent', TabHandles{prompt\_page,1},...

'String', KWHR\_battery\_size,'Callback', @battery\_size\_next\_pop, 'tag', 'KW\_menu', 'Visible', 'OFF', 'FontSize', 20);

% Create button for next click

battery\_size\_next\_button = uicontrol('Units', 'normalized', 'Position',[0.35 0.4 0.3 0.1], 'Style', 'pushbutton',...

'String', 'Next','Visible', 'On','Callback', @battery\_cost,'Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20, 'Visible', 'OFF');

% Create persistent variables

persistent battery\_size\_input;

battery\_size\_input = 0;

% Create function for click

function battery\_size\_next\_pop(hObject, eventdata)

% Create a next button once an option has been selected

set(battery\_size\_next\_button,'Visible','On')

% Find the answer and set output

index = get(hObject, 'Value');

battery\_size\_input = KWHR\_battery\_size(index)

set(battery\_size\_value, 'String', num2str(KWHR\_battery\_size(index)))

end

%% Cost of Battery

% Create the related questions

cost\_battery\_question = uicontrol('Units', 'normalized', 'Position',standard\_question, 'Style', 'text',...

'String', 'How much did your BES system cost ($)?','Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20, 'Visible', 'OFF');

% Create persistent variables

persistent cost\_battery\_input;

cost\_battery\_input = 0;

% Create function selection click

function battery\_cost(hObject, eventdata)

% Create a next button once an option has been selected

set(battery\_size\_question,'Visible','OFF')

set(KWHR\_popupmenu,'Visible','OFF')

set(battery\_size\_next\_button,'Visible','OFF')

% Find the answer and set output

set(cost\_battery\_question,'Visible','ON')

set(battery\_cost\_popupmenu,'Visible','ON')

progress\_bar(6.25);

end

% Dropdown list

battery\_cost\_list = [3000 4000 5000 6000 7000 8000 9000 10000];

% Set up pop up menu with pulldown data

battery\_cost\_popupmenu = uicontrol('Units', 'normalized', 'Position', [0.35 0.5 0.3 0.15], 'Style', 'popupmenu','Parent', TabHandles{prompt\_page,1},...

'String', battery\_cost\_list,'Callback', @cost\_bat\_next\_button, 'tag', 'KW\_menu', 'Visible', 'OFF', 'FontSize', 20);

% Create function for click

function cost\_bat\_next\_button(hObject, eventdata)

% Create a next button once an option has been selected

set(battery\_cost\_next\_button,'Visible','ON')

% Find the answer and set output

index = get(hObject, 'Value')

cost\_battery\_input = battery\_cost\_list(index)

set(battery\_cost\_value, 'String', num2str(battery\_cost\_list(index)))

progress\_bar(6.75);

end

% Create button for next click

battery\_cost\_next\_button = uicontrol('Units', 'normalized', 'Position',[0.35 0.4 0.3 0.1], 'Style', 'pushbutton',...

'String', 'Next', 'Visible', 'Off','Callback', @battery\_click\_no,'Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20);

% Create function for click

function battery\_click\_no(hObject, eventdata)

% Set previous questions off and next on with progress

set(text\_battery\_question,'Visible','OFF')

set(button\_yes\_battery,'Visible','OFF')

set(button\_no\_battery,'Visible','OFF')

set(cost\_battery\_question,'Visible','Off')

set(battery\_cost\_popupmenu,'Visible','Off')

set(battery\_cost\_next\_button,'Visible','Off')

set(text\_roof\_question,'Visible','ON')

set(tilt\_popupmenu,'Visible','ON')

progress\_bar(7);

end

%% Roof Tilt

% Create the related questions

text\_roof\_question = uicontrol('Units', 'normalized', 'Position',standard\_question, 'Style', 'text',...

'String', 'What is the angle of your roof (Degrees)?', 'Visible', 'On','Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20, 'Visible', 'OFF');

% Create persistent variables

persistent roof\_tilt\_input

% Create function selection click

function roof\_next(hObject, eventdata)

% Create a next button once an option has been selected

set(roof\_next\_button,'Visible','ON')

% Find the answer and set output

index = get(hObject, 'Value');

roof\_tilt\_input = roof\_tilt(index) ;

set(tilt\_value,'string', num2str(roof\_tilt(index)))

end

% Dropdown list

roof\_tilt = [0 5 10 15 20 25 30 35 40 45];

% Set up pop up menu with pulldown data

tilt\_popupmenu = uicontrol('Units', 'normalized', 'Position', [0.35 0.5 0.3 0.15], 'Style', 'popupmenu','Parent', TabHandles{prompt\_page,1},...

'String', roof\_tilt,'Callback', @roof\_next, 'tag', 'KW\_menu', 'Visible', 'OFF', 'FontSize', 20);

% Create button for next click

roof\_next\_button = uicontrol('Units', 'normalized', 'Position',[0.35 0.4 0.3 0.1], 'Style', 'pushbutton',...

'String', 'Next','Visible', 'On','Callback', @roof\_click,'Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20, 'Visible', 'OFF');

% Create function for click

function roof\_click(hObject, eventdata)

% Set previous questions off and next on with progress

set(text\_roof\_question,'Visible','OFF')

set(tilt\_popupmenu,'Visible','OFF')

set(roof\_next\_button,'Visible','OFF')

set(text\_orientation\_question,'Visible','ON')

set(orientation\_next\_button,'Visible','On')

set(orientation\_edit\_display,'Visible','On')

set(text\_orientation\_question,'Visible','On')

set(compass\_image,'Visible','On')

set(radio\_north\_button,'Visible','On'); set(radio\_north\_west\_button,'Visible','On')

set(radio\_east\_button,'Visible','On'); set(radio\_north\_east\_button,'Visible','On')

set(radio\_south\_button,'Visible','On'); set(radio\_south\_east\_button,'Visible','On')

set(radio\_west\_button,'Visible','On'); set(radio\_south\_west\_button,'Visible','On')

progress\_bar(8);

end

%% Roof Orientation

% Create the related questions

text\_orientation\_question = uicontrol('Units', 'normalized', 'Position',standard\_question, 'Style', 'text',...

'String', 'What orientation is your roof?', 'Visible', 'Off','Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20);

% Load Compass Image

[x,map]=imread('compass.jpg'); I2=imresize(x, [280 300]);

compass\_image = uicontrol('style','pushbutton','units','normalized','position',[0.333 0.13 0.33 0.55],...

'cdata',I2, 'Visible', 'Off','Parent', TabHandles{prompt\_page,1});

% Orientations Major

radio\_north\_button = uicontrol('Units', 'normalized', 'Position',[0.52 0.57 0.02 0.05], 'Style', 'radio',...

'Backgroundcolor', 'white', 'FontSize', 20, 'Visible', 'off','callback', @orientation\_click,'Parent', TabHandles{prompt\_page,1},...

'tag','N');

radio\_south\_button = uicontrol('Units', 'normalized', 'Position',[0.52 0.19 0.02 0.05], 'Style', 'radio',...

'Backgroundcolor', 'white', 'FontSize', 20, 'Visible', 'off','callback', @orientation\_click,'Parent', TabHandles{prompt\_page,1},...

'tag','S');

radio\_west\_button = uicontrol('Units', 'normalized', 'Position',[0.38 0.33 0.02 0.05], 'Style', 'radio',...

'Backgroundcolor', 'white', 'FontSize', 20, 'Visible', 'off','callback', @orientation\_click,'Parent', TabHandles{prompt\_page,1},...

'tag','W');

radio\_east\_button = uicontrol('Units', 'normalized', 'Position',[0.6 0.33 0.02 0.05], 'Style', 'radio',...

'Backgroundcolor', 'white', 'FontSize', 20, 'Visible', 'off','callback', @orientation\_click,'Parent', TabHandles{prompt\_page,1},...

'tag','E');

% Orientations Minor

radio\_north\_west\_button = uicontrol('Units', 'normalized', 'Position',[0.39 0.57 0.02 0.05], 'Style', 'radio',...

'Backgroundcolor', 'white', 'FontSize', 20, 'Visible', 'off','callback', @orientation\_click,'Parent', TabHandles{prompt\_page,1},...

'tag','NW');

radio\_north\_east\_button = uicontrol('Units', 'normalized', 'Position',[0.58 0.57 0.02 0.05], 'Style', 'radio',...

'Backgroundcolor', 'white', 'FontSize', 20, 'Visible', 'off','callback', @orientation\_click,'Parent', TabHandles{prompt\_page,1},...

'tag','NE');

radio\_south\_west\_button = uicontrol('Units', 'normalized', 'Position',[0.39 0.19 0.02 0.05], 'Style', 'radio',...

'Backgroundcolor', 'white', 'FontSize', 20, 'Visible', 'off','callback', @orientation\_click,'Parent', TabHandles{prompt\_page,1},...

'tag','SW');

radio\_south\_east\_button = uicontrol('Units', 'normalized', 'Position',[0.58 0.19 0.02 0.05], 'Style', 'radio',...

'Backgroundcolor', 'white', 'FontSize', 20, 'Visible', 'off','callback', @orientation\_click,'Parent', TabHandles{prompt\_page,1},...

'tag','SE');

% Create button for next click

orientation\_next\_button = uicontrol('Units', 'normalized', 'Position',[0.67 0.26 0.3 0.1], 'Style', 'pushbutton',...

'String', 'Next', 'Visible', 'Off','Callback', @orientation\_next,'Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 20);

% Create button for display

orientation\_edit\_display = uicontrol('Units', 'normalized', 'Position',[0.67 0.37 0.3 0.1], 'Style', 'text',...

'tag','orientation\_selection', 'Visible', 'Off','Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', 'white', 'Foregroundcolor', 'black', 'FontSize', 20);

% Create function for entry

function orientation\_click(hObject, eventdata)

% Create a next button once an option has been selected

set(orientation\_next\_button,'Visible','On')

set(orientation\_edit\_display,'Visible','On')

set(text\_orientation\_question,'Visible','On')

% Find the answer and set output

string = get(hObject, 'tag');

% Find which popupmenu was selected and update the variable display box

if strcmp(string, 'N')

set(orientation\_edit\_display, 'String', 'North'); orientation\_input = 1

set(orientation\_value, 'String', 'North');

elseif strcmp(string, 'S')

set(orientation\_edit\_display, 'String', 'South') ; orientation\_input = 3

elseif strcmp(string, 'E')

set(orientation\_edit\_display, 'String', 'East') ; orientation\_input = 2

elseif strcmp(string, 'W')

set(orientation\_edit\_display, 'String', 'West') ; orientation\_input = 4

elseif strcmp(string, 'NE')

set(orientation\_edit\_display, 'String', 'North-East') ; orientation\_input = 5

set(orientation\_value, 'String', 'North-East') ;

elseif strcmp(string, 'NW')

set(orientation\_edit\_display, 'String', 'North-West') ; orientation\_input = 8

set(orientation\_value, 'String', 'North-West') ;

elseif strcmp(string, 'SE')

set(orientation\_edit\_display, 'String', 'South-East') ; orientation\_input = 6

elseif strcmp(string, 'SW')

set(orientation\_edit\_display, 'String', 'South-West'); orientation\_input = 7

end

% If a selection other than North is chosen display an error

if orientation\_input ~=1

set(orientation\_next\_button,'Visible','Off')

errordlg('Error only Northern facing arrays','Setup Error')

return

end

end

% Create function for battery question

function orientation\_next(hObject, eventdata)

% Set previous questions off and next on with progress

set(orientation\_next\_button,'Visible','Off')

set(orientation\_edit\_display,'Visible','Off')

set(text\_orientation\_question,'Visible','Off')

set(compass\_image,'Visible','Off')

set(radio\_north\_button,'Visible','Off'); set(radio\_north\_west\_button,'Visible','Off')

set(radio\_east\_button,'Visible','Off'); set(radio\_north\_east\_button,'Visible','Off')

set(radio\_south\_button,'Visible','Off'); set(radio\_south\_east\_button,'Visible','Off')

set(radio\_west\_button,'Visible','Off'); set(radio\_south\_west\_button,'Visible','Off')

set(text\_state\_question,'Visible','On');

set(postcode\_edit,'Visible','On');

set(text\_state\_codes,'Visible','On');

set(post\_code\_table,'Visible','On');

progress\_bar(9);

end

%% Post Code

% Create the related questions

text\_state\_question = uicontrol('Units', 'normalized', 'Position',standard\_question, 'Style', 'text',...

'String', 'What is your post code?', 'Visible', 'Off','Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20);

text\_state\_codes = uicontrol('Units', 'normalized', 'Position',[0.2 0.35 0.1 0.3], 'Style', 'text',...

'String', '4814 TSV 4825 ISA 0800 DAR 6000 PER 3000 MEL 7000 HOB 2000 SYD 4000 BRI', 'Visible', 'Off','Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 15);

% Create persistent variables

persistent state\_input

% Dropdown list for post codes

state\_codes = [4814 4825 0800 6000 3000 7000 2000 4000];

% List of names of post codes

state\_names = {'Townsville, QLD'; 'Mount Isa, QLD'; 'Darwin, NT'; 'Perth, WA'; 'Melbourne, VIC';...

'Horbart, TAS'; 'Sydney, NSW'; 'Brisbane, QLD'};

% Create button for edit

postcode\_edit = uicontrol('Units', 'normalized', 'Position', [0.35 0.5 0.3 0.1], 'Style', 'edit','Parent', TabHandles{prompt\_page,1},...

'Callback', @state\_check\_next, 'tag', 'state\_entry', 'Visible', 'Off', 'FontSize', 20);

% Create table to display postcode options

state\_labels =[num2cell(state\_codes') state\_names] ;

columnname = {'Postcode', 'Location'};

columnformat = {'char', 'char'};

columneditable = [false false];

post\_code\_table = uitable('Units','normalized','Position',[0.15 0.35 0.18 0.3],'Parent', TabHandles{prompt\_page,1},...

'Data', state\_labels, 'Visible', 'Off',...

'ColumnName', columnname,...

'ColumnFormat', columnformat,...

'ColumnEditable', columneditable,...

'RowName',[] ,'BackgroundColor',[.7 .9 .8],...

'ForegroundColor',[0 0 0],'ColumnWidth',{'auto' 140},'FontSize',14);

% Create function for click

function state\_check\_next(hObject, eventdata)

% Computes the state code

% Displays the post code

state\_input = str2double(get(postcode\_edit,'string'));

set(state\_display\_button,'Visible','ON')

set(state\_display\_button, 'String', "Invalid")

% Checks the post code against all options for errors

for i = 1:1:length(state\_codes)

if state\_input == state\_codes(i)

set(state\_display\_button, 'String', state\_names(i))

set(state\_next\_button,'Visible','ON')

set(state\_value, 'String', state\_names(i))

set(postal\_value, 'String', num2str(state\_input))

set(tariff\_value, 'String', '11')

set(supplier\_value, 'String', 'Ergon')

progress\_bar(10);

% Displays error for no gas data

if state\_input == 4814||4825||0800

errordlg('Error - We do not have benchmark data for mains gas for this postcode,... data without mains gas connected is shown instead.','Data Error')

return

end

% Displays error for no gas data

if state\_input == 6000

errordlg('Error - Your estimate will be based on WA consumption data from 2011, which is the most recent... available to the AER. It assumes you have no pool and use an average amount of gas.','Data Error')

return

end

end

end

end

% Create display of state to confirm

state\_display\_button = uicontrol('Units', 'normalized', 'Position',[0.35 0.4 0.3 0.1], 'Style', 'text',...

'String', 'Location','Visible', 'Off','Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', 'green', 'Foregroundcolor', 'black', 'FontSize', 20);

% Create button for next

state\_next\_button = uicontrol('Units', 'normalized', 'Position',[0.35 0.2 0.3 0.1], 'Style', 'pushbutton',...

'String', 'Next','Visible', 'Off','Callback', @state\_click,'Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20);

% Create function for end of state codes

function state\_click(hObject, eventdata)

% Set previous questions off and next on with progress

set(state\_display\_button,'Visible','Off')

set(state\_next\_button,'Visible','Off')

set(text\_state\_question,'Visible','Off')

set(postcode\_edit,'Visible','Off')

set(text\_state\_codes,'Visible','Off');

set(post\_code\_table,'Visible','Off')

set(text\_bill\_question,'Visible','On')

set(bill\_edit,'Visible','On')

set(bill\_skip\_button,'Visible','On')

progress\_bar(11);

end

%% Bill Entry

% Create the related questions

text\_bill\_question = uicontrol('Units', 'normalized', 'Position',standard\_question, 'Style', 'text',...

'String', 'Cost of last quarter bill ($)?', 'Visible', 'Off','Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20);

% Create button for display

bill\_edit = uicontrol('Units', 'normalized', 'Position', [0.35 0.5 0.3 0.1], 'Style', 'edit','Parent', TabHandles{prompt\_page,1},...

'Callback', @bill\_next, 'tag', 'state\_entry', 'Visible', 'Off', 'FontSize', 20);

% Create persistent variables

persistent bill\_input

bill\_input=0;

% Create function selection click

function bill\_next(hObject, eventdata)

% Create a next button once an option has been selected

bill\_input = str2double(get(bill\_edit,'string'))

set(bill\_skip\_button,'Visible','Off')

set(bill\_next\_button,'Visible','ON')

set(bill\_value, 'String', get(bill\_edit,'string'))

end

% Create button for bill skip

bill\_skip\_button = uicontrol('Units', 'normalized', 'Position',[0.35 0.4 0.3 0.1], 'Style', 'pushbutton',...

'String', 'Skip','Visible', 'Off','Callback', @bill\_click,'Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20);

% Create button for next

bill\_next\_button = uicontrol('Units', 'normalized', 'Position',[0.35 0.4 0.3 0.1], 'Style', 'pushbutton',...

'String', 'Next','Visible', 'Off','Callback', @bill\_click,'Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20);

% Create function for click

function bill\_click(hObject, eventdata)

% Set previous questions off and next on with progress

set(text\_bill\_question,'Visible','Off')

set(bill\_edit,'Visible','Off')

set(bill\_next\_button,'Visible','Off')

set(bill\_skip\_button,'Visible','Off')

set(number\_people\_question,'Visible','On')

set(people\_popupmenu,'Visible','On')

progress\_bar(12);

end

%% Number of People

% Create the related questions

number\_people\_question = uicontrol('Units', 'normalized', 'Position',standard\_question, 'Style', 'text',...

'String', 'How many occupants are in the residence?', 'Visible', 'Off','Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20);

% Create persistent variables

persistent number\_people\_input;

% Create function selection click

function display\_people\_next\_button(hObject, eventdata)

% Create a next button once an option has been selected

set(people\_next\_button,'Visible','ON')

% Find the answer and set output

index = get(hObject, 'Value');

number\_people\_input = number\_people(index)

set(occupants\_value, 'String', num2str(number\_people(index)))

end

% Dropdown list

number\_people = [1 2 3 4];

% Set up pop up menu with pulldown data

people\_popupmenu = uicontrol('Units', 'normalized', 'Position', [0.35 0.5 0.3 0.15], 'Style', 'popupmenu','Parent', TabHandles{prompt\_page,1},...

'String', number\_people,'Callback', @display\_people\_next\_button, 'tag', 'KW\_menu', 'Visible', 'OFF', 'FontSize', 20);

% Create button for next click

people\_next\_button = uicontrol('Units', 'normalized', 'Position',[0.35 0.4 0.3 0.1], 'Style', 'pushbutton',...

'String', 'Next', 'Visible', 'Off','Callback', @people\_click,'Parent', TabHandles{prompt\_page,1},...

'Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 20);

% Create function for click

function people\_click(hObject, eventdata)

% Set the number of people

set(number\_people\_question,'Visible','Off')

set(people\_popupmenu,'Visible','Off')

set(people\_next\_button,'Visible','Off')

% Run the main function to calculate inputs

PSH\_and\_KW\_Calc(solar\_size\_input,performance\_input, roof\_tilt\_input,state\_input...

,number\_people\_input,gas\_mains\_input, pool\_input, battery\_size\_input, solar\_installed, battery\_installed)

% Display the input tab, update progress and turn on the enter prompt

TabSellectCallback(0,0,2);

set(enter\_gui\_button,'Visible','On')

progress\_bar(13);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Tab 2 Content: INPUTS TAB %%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Create the UI Handles for the labels and Values on Input Page

% Create tab reference for parent handles

y\_offset = -0.05;

% Set up page number for referencing

input\_page = 2;

% Create the Labels for each value

solar\_title = uicontrol('Units', 'normalized', 'Position',[0.3 0.9+y\_offset 0.4 0.075], 'Style', 'text','Parent', TabHandles{input\_page,1},...

'String', 'Solar Parameter Inputs', 'Visible', 'On','Backgroundcolor',[0.5 1 0], 'Foregroundcolor', 'black', 'FontSize', 20);

current\_system = uicontrol('Units', 'normalized', 'Position',[0.1 0.8+y\_offset 0.15 0.05], 'Style', 'text','Parent', TabHandles{input\_page,1},...

'String', 'Location', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 10);

postal\_title = uicontrol('Units', 'normalized', 'Position',[0.1 0.7+y\_offset 0.15 0.05], 'Style', 'text','Parent', TabHandles{input\_page,1},...

'String', 'Post Code', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 10);

bill\_title = uicontrol('Units', 'normalized', 'Position',[0.1 0.6+y\_offset 0.15 0.05], 'Style', 'text','Parent', TabHandles{input\_page,1},...

'String', 'Bill $', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 10);

supplier\_title = uicontrol('Units', 'normalized', 'Position',[0.1 0.5+y\_offset 0.15 0.05], 'Style', 'text','Parent', TabHandles{input\_page,1},...

'String', 'Supplier', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 10);

tariff\_title = uicontrol('Units', 'normalized', 'Position',[0.1 0.4+y\_offset 0.15 0.05], 'Style', 'text','Parent', TabHandles{input\_page,1},...

'String', 'Tariff', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 10);

occupants\_title = uicontrol('Units', 'normalized', 'Position',[0.1 0.3+y\_offset 0.15 0.05], 'Style', 'text','Parent', TabHandles{input\_page,1},...

'String', '# Occupants', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 10);

gas\_main\_title = uicontrol('Units', 'normalized', 'Position',[0.1 0.2+y\_offset 0.15 0.05], 'Style', 'text','Parent', TabHandles{input\_page,1},...

'String', 'Gas Mains', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 10);

pool\_title = uicontrol('Units', 'normalized', 'Position',[0.1 0.1+y\_offset 0.15 0.05], 'Style', 'text','Parent', TabHandles{input\_page,1},...

'String', 'Pool Connected', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 10);

% Create the edit boxes to update

state\_value = uicontrol('Units', 'normalized', 'Position',[0.3 0.8+y\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{input\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

postal\_value = uicontrol('Units', 'normalized', 'Position',[0.3 0.7+y\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{input\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

bill\_value = uicontrol('Units', 'normalized', 'Position',[0.3 0.6+y\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{input\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

supplier\_value = uicontrol('Units', 'normalized', 'Position',[0.3 0.5+y\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{input\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

tariff\_value = uicontrol('Units', 'normalized', 'Position',[0.3 0.4+y\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{input\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

occupants\_value = uicontrol('Units', 'normalized', 'Position',[0.3 0.3+y\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{input\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

gas\_main\_value = uicontrol('Units', 'normalized', 'Position',[0.3 0.2+y\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{input\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

pool\_value = uicontrol('Units', 'normalized', 'Position',[0.3 0.1+y\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{input\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

% Create the Labels for each value

system\_title = uicontrol('Units', 'normalized', 'Position',[0.65 0.8+y\_offset 0.15 0.05], 'Style', 'text','Parent', TabHandles{input\_page,1},...

'String', 'System Specifications', 'Visible', 'On','Backgroundcolor', 'green', 'Foregroundcolor', 'black', 'FontSize', 10);

solar\_size\_title = uicontrol('Units', 'normalized', 'Position',[0.55 0.7+y\_offset 0.15 0.05], 'Style', 'text','Parent', TabHandles{input\_page,1},...

'String', 'Solar Size (kW)', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 10);

solar\_cost\_title = uicontrol('Units', 'normalized', 'Position',[0.55 0.6+y\_offset 0.15 0.05], 'Style', 'text','Parent', TabHandles{input\_page,1},...

'String', 'Solar Cost $', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 10);

battery\_size\_title = uicontrol('Units', 'normalized', 'Position',[0.55 0.5+y\_offset 0.15 0.05], 'Style', 'text','Parent', TabHandles{input\_page,1},...

'String', 'Battery Size (kWhr)', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 10);

battery\_cost\_title = uicontrol('Units', 'normalized', 'Position',[0.55 0.4+y\_offset 0.15 0.05], 'Style', 'text','Parent', TabHandles{input\_page,1},...

'String', ' Battery Cost $', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 10);

tilt\_title = uicontrol('Units', 'normalized', 'Position',[0.55 0.2+y\_offset 0.15 0.05], 'Style', 'text','Parent', TabHandles{input\_page,1},...

'String', 'Roof Tilt (Degrees)', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 10);

orientation\_title = uicontrol('Units', 'normalized', 'Position',[0.55 0.1+y\_offset 0.15 0.05], 'Style', 'text','Parent', TabHandles{input\_page,1},...

'String', 'Roof Orientation', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 10);

% Create the edit boxes to update

solar\_size\_value = uicontrol('Units', 'normalized', 'Position',[0.75 0.7+y\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{input\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

solar\_cost\_value = uicontrol('Units', 'normalized', 'Position',[0.75 0.6+y\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{input\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

battery\_size\_value = uicontrol('Units', 'normalized', 'Position',[0.75 0.5+y\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{input\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

battery\_cost\_value = uicontrol('Units', 'normalized', 'Position',[0.75 0.4+y\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{input\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

roofspec\_title = uicontrol('Units', 'normalized', 'Position',[0.65 0.3+y\_offset 0.15 0.05], 'Style', 'text','Parent', TabHandles{input\_page,1},...

'String', 'Roof Specifications', 'Visible', 'On','Backgroundcolor', 'green', 'Foregroundcolor', 'black', 'FontSize', 10);

tilt\_value = uicontrol('Units', 'normalized', 'Position',[0.75 0.2+y\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{input\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

orientation\_value = uicontrol('Units', 'normalized', 'Position',[0.75 0.1+y\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{input\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Tab 3 Content: PRODUCTION TAB %%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Create the UI Handles for the labels and Values on Production Page

% Create tab reference for parent handles

y\_prod\_offset = 0.025;

% Set up page number for referencing

production\_page = 3;

% Create the Labels for each value

current\_system = uicontrol('Units', 'normalized', 'Position',[0.1 0.875+y\_prod\_offset 0.35 0.07], 'Style', 'text','Parent', TabHandles{production\_page,1},...

'String', 'Estimated Daily Production (kWhr)', 'Visible', 'On','Backgroundcolor',[0.5 1 0], 'Foregroundcolor', 'black', 'FontSize', 20);

current\_system = uicontrol('Units', 'normalized', 'Position',[0.1 0.8+y\_prod\_offset 0.175 0.05], 'Style', 'text','Parent', TabHandles{production\_page,1},...

'String', 'Daily Usuage', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 15);

current\_system = uicontrol('Units', 'normalized', 'Position',[0.1 0.725+y\_prod\_offset 0.175 0.05], 'Style', 'text','Parent', TabHandles{production\_page,1},...

'String', 'Daily Solar Production', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 15);

current\_system = uicontrol('Units', 'normalized', 'Position',[0.1 0.65+y\_prod\_offset 0.175 0.05], 'Style', 'text','Parent', TabHandles{production\_page,1},...

'String', 'Daily Storage', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 15);

current\_system = uicontrol('Units', 'normalized', 'Position',[0.1 0.575+y\_prod\_offset 0.175 0.05], 'Style', 'text','Parent', TabHandles{production\_page,1},...

'String', 'Total Exported', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 15);

% Create the edit boxes to update

daily\_usuage\_value = uicontrol('Units', 'normalized', 'Position',[0.3 0.8+y\_prod\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{production\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

daily\_production\_value = uicontrol('Units', 'normalized', 'Position',[0.3 0.725+y\_prod\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{production\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

daily\_storage\_value = uicontrol('Units', 'normalized', 'Position',[0.3 0.65+y\_prod\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{production\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

daily\_exported\_value = uicontrol('Units', 'normalized', 'Position',[0.3 0.575+y\_prod\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{production\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

% Create the Labels for each value

current\_system = uicontrol('Units', 'normalized', 'Position',[0.575 0.875+y\_prod\_offset 0.375 0.07], 'Style', 'text','Parent', TabHandles{production\_page,1},...

'String', 'Daily Cost ($)', 'Visible', 'On','Backgroundcolor',[0.5 1 0], 'Foregroundcolor', 'black', 'FontSize', 20);

daily\_normal\_cost\_title = uicontrol('Units', 'normalized', 'Position',[0.575 0.8+y\_prod\_offset 0.2 0.05], 'Style', 'text','Parent', TabHandles{production\_page,1},...

'String', 'Standard', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 15);

daily\_solar\_cost\_title = uicontrol('Units', 'normalized', 'Position',[0.575 0.725+y\_prod\_offset 0.2 0.05], 'Style', 'text','Parent', TabHandles{production\_page,1},...

'String', 'Import', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 15);

current\_system = uicontrol('Units', 'normalized', 'Position',[0.575 0.65+y\_prod\_offset 0.2 0.05], 'Style', 'text','Parent', TabHandles{production\_page,1},...

'String', 'Export', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 15);

current\_system = uicontrol('Units', 'normalized', 'Position',[0.575 0.575+y\_prod\_offset 0.2 0.05], 'Style', 'text','Parent', TabHandles{production\_page,1},...

'String', 'Actual Savings', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 15);

% Create the edit boxes to update

daily\_normal\_cost\_value = uicontrol('Units', 'normalized', 'Position',[0.8 0.8+y\_prod\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{production\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

daily\_import\_cost\_value = uicontrol('Units', 'normalized', 'Position',[0.8 0.725+y\_prod\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{production\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

daily\_export\_cost\_value = uicontrol('Units', 'normalized', 'Position',[0.8 0.65+y\_prod\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{production\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

daily\_savings\_cost\_value = uicontrol('Units', 'normalized', 'Position',[0.8 0.575+y\_prod\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{production\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

% Calculation of PSH from tilt angle

% Create variables to be seen by all functions

persistent kw\_produced\_daily

persistent daily\_savings

persistent source\_energy

% This is the main code used in program

function PSH\_and\_KW\_Calc(solar\_size\_input,performance\_input, roof\_tilt\_input,state\_input...

,number\_people\_input,gas\_mains\_input, pool\_input, battery\_size\_input, solar\_installed, battery\_installed)

% This functions uses the imported saved data

PSH\_avg = tilt\_calculator (roof\_tilt\_input,13);

% disp('Peak Sun Hours');

% disp(PSH\_avg);

% This determines if the used input a bill value

if bill\_input == 0

kwhr\_avg\_found = average\_kwhr\_finder(state\_input,number\_people\_input,gas\_mains\_input, pool\_input) ;

else

kwhr\_avg\_found = average\_kwhr\_finder(state\_input,number\_people\_input,gas\_mains\_input, pool\_input) ;

kwhr\_avg\_found = bill\_input/(90\*tariff\_rate\_normal\_found)

end

% Displays the average kilowatts for household

% disp('Average Kilowatts Hours For Household');

% disp(kwhr\_avg\_found);

% This determines how much is produced daily

kw\_produced\_daily = solar\_size\_input \* PSH\_avg \* performance\_input;

kwhr\_used\_from\_solar = (10/percentage\_input)\*kw\_produced\_daily;

% Calculates Daily storage

daily\_storage = battery\_size\_input;

if ((battery\_installed == 1) & (solar\_installed == 0))

daily\_exported = 0;

elseif ((battery\_installed == 0) & (solar\_installed == 0))

kw\_produced\_daily = 0;

else

daily\_exported = (kw\_produced\_daily - kwhr\_used\_from\_solar - daily\_storage);

end

% Calculates how much is imported from the grid

daily\_imported = (kwhr\_avg\_found - kwhr\_used\_from\_solar - daily\_storage);

daily\_savings = (kwhr\_avg\_found\*tariff\_rate\_normal\_found -...

(daily\_imported\*tariff\_rate\_normal\_found - daily\_exported\*solar\_rate\_feedin\_found));

% Updates the values to the screen for the user

Update\_Values\_prod(kwhr\_avg\_found,kw\_produced\_daily,daily\_storage,tariff\_rate\_normal\_found,daily\_exported,daily\_imported,daily\_savings)

% Runs the functions for cost and plots the values

production\_graph()

cost\_analysis(daily\_savings)

finance\_graph()

% Protects the pie graph against non-positive data

if (daily\_storage <= 0)

pie\_storage = 0.00000000001;

else

pie\_storage = daily\_storage;

end

if (daily\_imported <= 0)

daily\_imported\_pie = 0.00000000001;

else

daily\_imported\_pie = daily\_imported;

end

% Code for the Pie Chart creations

source\_energy = [kwhr\_used\_from\_solar pie\_storage daily\_imported\_pie]/...

(kwhr\_used\_from\_solar + pie\_storage + daily\_imported\_pie);

disp(source\_energy(1,1));disp(source\_energy(1,2));disp(source\_energy(1,3));

% Pie chart axis

haxes\_pie = axes('Parent', TabHandles{display\_page,1}, ...

'Units', 'normalized', ...

'Position', [0.7 0.665 0.3 0.3]);

% This code is to name the pie chart and name the percentages

pie\_face = pie(haxes\_pie,source\_energy);

title(haxes\_pie,'Daily Energy Sources');

jooda = pie\_face(1); jooda.FaceColor = 'green'; jooda = pie\_face(2); jooda .FontSize = 12;

jooda = pie\_face(3); jooda.FaceColor = 'yellow'; jooda = pie\_face(4); jooda .FontSize = 10;

jooda = pie\_face(5); jooda.FaceColor = 'red'; jooda = pie\_face(6); jooda .FontSize = 10;

hText = findobj(pie\_face,'Type','text'); % text object handles

percentValues = get(hText,'String'); % percent values

energy\_sources = {'Solar offset: ';'Battery offset: ';'Grid Imports: '};

combinedtxt = strcat(energy\_sources,percentValues); % strings and percent values

oldExtents\_cell = get(hText,'Extent'); % cell array

oldExtents = cell2mat(oldExtents\_cell); % numeric array

hText(1).String = combinedtxt(1);

hText(2).String = combinedtxt(2);

hText(3).String = combinedtxt(3);

newExtents\_cell = get(hText,'Extent'); % cell array

newExtents = cell2mat(newExtents\_cell); % numeric array

width\_change = newExtents(:,3)-oldExtents(:,3);

signValues = sign(oldExtents(:,1));

offset = signValues.\*(width\_change/2);

textPositions\_cell = get(hText,{'Position'}); % cell array

textPositions = cell2mat(textPositions\_cell); % numeric array

textPositions(:,1) = textPositions(:,1) + offset; % add offset

hText(1).Position = textPositions(1,:);

hText(2).Position = textPositions(2,:);

hText(3).Position = textPositions(3,:);

% This was the larger pie chart with the same values but on

% another tab

haxes\_pie8 = axes('Parent', TabHandles{8,1}, ...

'Units', 'normalized', ...

'Position', [0.05 0.1 0.9 0.75]);

% This code is to name the pie chart and name the percentages

pie\_face8 = pie(haxes\_pie8,source\_energy);

title(haxes\_pie8,'Daily Energy Sources');

set(gca,'fontsize',20)

jooda = pie\_face8(1); jooda.FaceColor = 'green'; jooda = pie\_face8(2); jooda .FontSize = 16;

jooda = pie\_face8(3); jooda.FaceColor = 'yellow'; jooda = pie\_face8(4); jooda .FontSize = 16;

jooda = pie\_face8(5); jooda.FaceColor = 'red'; jooda = pie\_face8(6); jooda .FontSize = 16;

hText = findobj(pie\_face8,'Type','text'); % text object handles

percentValues = get(hText,'String'); % percent values

energy\_sources = {'Solar offset: ';'Battery offset: ';'Grid Imports: '};

combinedtxt = strcat(energy\_sources,percentValues); % strings and percent values

oldExtents\_cell = get(hText,'Extent'); % cell array

oldExtents = cell2mat(oldExtents\_cell); % numeric array

hText(1).String = combinedtxt(1);

hText(2).String = combinedtxt(2);

hText(3).String = combinedtxt(3);

newExtents\_cell = get(hText,'Extent'); % cell array

newExtents = cell2mat(newExtents\_cell); % numeric array

width\_change = newExtents(:,3)-oldExtents(:,3);

signValues = sign(oldExtents(:,1));

offset = signValues.\*(width\_change/2);

textPositions\_cell = get(hText,{'Position'}); % cell array

textPositions = cell2mat(textPositions\_cell); % numeric array

textPositions(:,1) = textPositions(:,1) + offset; % add offset

hText(1).Position = textPositions(1,:);

hText(2).Position = textPositions(2,:);

hText(3).Position = textPositions(3,:);

end

% Function for finding the linear change between tilt angle and NASA

% PSH

function [PSH\_avg] = tilt\_calculator (roof\_tilt\_input,month)

% Create variables for rows and columns

psh\_row\_index = 0;

month = month + 1;

% Switch statements determines the post code

switch state\_input

case 4814

case 4825

psh\_row\_index = 5;

case 0800

psh\_row\_index = 10;

case 6000

psh\_row\_index = 15;

case 3000

psh\_row\_index = 20;

case 7000

psh\_row\_index = 25;

case 2000

psh\_row\_index = 30;

case 4000

psh\_row\_index = 35;

end

% Data creates the min and max values

max\_tilt0 = solar\_psh\_data(psh\_row\_index+1,1);

max\_tilt1 = solar\_psh\_data(psh\_row\_index+2,1);

max\_tilt2 = solar\_psh\_data(psh\_row\_index+3,1);

max\_tilt3 = solar\_psh\_data(psh\_row\_index+4,1);

max\_tilt4 = solar\_psh\_data(psh\_row\_index+5,1);

% If statements index the data for analysis

if ((roof\_tilt\_input >= max\_tilt0) & (roof\_tilt\_input <= max\_tilt1))

max\_tilt = max\_tilt1; min\_tilt = max\_tilt0;

max\_psh = solar\_psh\_data(2+psh\_row\_index,month); min\_psh = solar\_psh\_data(1+psh\_row\_index,month);

elseif ((roof\_tilt\_input > max\_tilt1) & (roof\_tilt\_input <= max\_tilt2))

max\_tilt = max\_tilt2; min\_tilt = max\_tilt1;

max\_psh = solar\_psh\_data(3+psh\_row\_index,month); min\_psh = solar\_psh\_data(2+psh\_row\_index,month);

elseif ((roof\_tilt\_input > max\_tilt2) & (roof\_tilt\_input <= max\_tilt3))

max\_tilt = max\_tilt3; min\_tilt = max\_tilt2;

max\_psh = solar\_psh\_data(4+psh\_row\_index,month); min\_psh = solar\_psh\_data(3+psh\_row\_index,month);

elseif ((roof\_tilt\_input> max\_tilt3) & (roof\_tilt\_input <= max\_tilt4))

max\_tilt = max\_tilt4; min\_tilt = max\_tilt3;

max\_psh = solar\_psh\_data(5+psh\_row\_index,month); min\_psh = solar\_psh\_data(4+psh\_row\_index,month);

end

% The PSH is calculated using the min and max tilts and a

% linear equation

div\_step = (max\_psh - min\_psh)/(max\_tilt-min\_tilt);

tilt\_offset = (roof\_tilt\_input - min\_tilt) \* div\_step ;

PSH\_avg = tilt\_offset + min\_psh;

end

% Function for plotting the production

function production\_graph()

% Create axis for graph

PlotOffset = 40;

haxes2 = axes('Parent', TabHandles{production\_page,1}, ...

'Units', 'normalized', ...

'Position', [0.075 0.1 0.9 0.45]);

% Create a cell array for the months to be plotted

month\_name = ({'Jan';'Feb';'Mar';'Apr';'May';'Jun';'Jul';'Aug';'Sep';'Oct';'Nov';'Dec';'Average'});

% Create a looop to plot the production agasint the months

for i = 1:1:13

kw\_produced\_daily\_plot = solar\_size\_input \* tilt\_calculator (roof\_tilt\_input,i) \* performance\_input;

bar\_data\_plot(1,i) = kw\_produced\_daily\_plot;

hold all

end

% Plot the data

bar(haxes2,bar\_data\_plot,'FaceColor',[0 .8 .5],'EdgeColor','yellow','LineWidth',1.5);

set(gca, 'XTick', 1:13,'xticklabel',month\_name)

% Label, Dimension and Legent the GRPAH

title('Monthly Average kWhr Production','Color','yellow','FontSize', 13);

xlabel('Month'); ylabel('Production (kWhr)');

set(gca, ...

'Box' , 'off' , ...

'TickDir' , 'out' , ...

'TickLength' , [.01 .01] , ...

'XMinorTick' , 'on' , ...

'YMinorTick' , 'on' , ...

'YGrid' , 'on' , ...

'XGrid' , 'off' , ...

'Color',Grey, 'FontSize', 13,...

'XColor' , 'yellow', ...

'YColor' , 'yellow', ...

'LineWidth' , 3 );

% End of Graph Labelling

% Create graph larger and on another tab for report

% Create another axis

haxes6 = axes('Parent', TabHandles{6,1}, ...

'Units', 'normalized', ...

'Position', [0.075 0.15 0.9 0.75]);

% Plot the data

bar(haxes6,bar\_data\_plot,'FaceColor',[0 .8 .5],'EdgeColor','yellow','LineWidth',1.5);

set(gca, 'XTick', 1:13,'xticklabel',month\_name)

% Label, Dimension and Legent the GRPAH

title('Monthly Average kWhr Production','Color','black','FontSize', 13);

xlabel('Month'); ylabel('Production (kWhr)');

ylim([0 30]);

% legend({'RAW Signal'});

set(gca, ...

'Box' , 'off' , ...

'TickDir' , 'out' , ...

'TickLength' , [.01 .01] , ...

'XMinorTick' , 'on' , ...

'YMinorTick' , 'on' , ...

'YGrid' , 'on' , ...

'XGrid' , 'off' , ...

'Color','white', 'FontSize', 13,...

'XColor' , 'black', ...

'YColor' , 'black', ...

'LineWidth' , 3 );

% End of Graph Labelling

end

% Calculates the tarriffs, and average kW usuage from post code

% Values for Taffifs

persistent tariff\_found

persistent tariff\_rate\_normal\_found

persistent solar\_rate\_feedin\_found

persistent kwhr\_avg\_found

% Main fucntion for indexing

function [kwhr\_avg\_found] = average\_kwhr\_finder(state\_input,number\_people\_input,gas\_mains\_input, pool\_input)

% Create variables for indexing rows and columns

column = 5;

row = 1;

% Switchstatement from state input

switch state\_input

case 4814

case 4825

row = row + 16;

case 0800

row = row + 32;

case 6000

row = row + 48;

case 3000

row = row + 64;

case 7000

row = row + 80;

case 2000

row = row + 96;

case 4000

row = row + 112;

end

% Switch statement for number of occupants

switch number\_people\_input

case 3

row = row +4;

case 2

row = row +8;

case 1

row = row +12;

end

% Switch statement for pool connection

switch pool\_input

case 1

row = row +2;

end

% Switch statement for gas mains

switch gas\_mains\_input

case 1

row = row +1;

end

% From the stored data the average kW is indexed

kwhr\_avg\_found = kwhr\_avg\_data(row,column);

tariff\_found = kwhr\_avg\_data(row,column+2);

tariff\_rate\_normal\_found = kwhr\_avg\_data(row,column+3);

solar\_rate\_feedin\_found = kwhr\_avg\_data(row,column+4);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Tab 4 Content: FINANCE TAB %%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Create the UI Handles for the labels and Values on Finance Page

% Create tab reference for parent handles

y\_finance\_offset = 0.025;

% Set up page number for referencing

finance\_page = 4;

% Create the Labels for each value

current\_system = uicontrol('Units', 'normalized', 'Position',[0.1 0.875+y\_prod\_offset 0.35 0.07], 'Style', 'text','Parent', TabHandles{finance\_page,1},...

'String', 'Finance Options ($/kWhr)', 'Visible', 'On','Backgroundcolor',[0.5 1 0], 'Foregroundcolor', 'black', 'FontSize', 20);

ALCC\_title = uicontrol('Units', 'normalized', 'Position',[0.1 0.8+y\_prod\_offset 0.175 0.05], 'Style', 'text','Parent', TabHandles{finance\_page,1},...

'String', 'ALCC $/kWhr', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 15);

ANNPMT\_opt\_title = uicontrol('Units', 'normalized', 'Position',[0.1 0.725+y\_prod\_offset 0.175 0.05], 'Style', 'text','Parent', TabHandles{finance\_page,1},...

'String', 'ANNPMT Optimistic', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 15);

ANNPMT\_likely\_title = uicontrol('Units', 'normalized', 'Position',[0.1 0.65+y\_prod\_offset 0.175 0.05], 'Style', 'text','Parent', TabHandles{finance\_page,1},...

'String', 'ANNPMT Likely', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 15);

ANNPMT\_pess\_title = uicontrol('Units', 'normalized', 'Position',[0.1 0.575+y\_prod\_offset 0.175 0.05], 'Style', 'text','Parent', TabHandles{finance\_page,1},...

'String', 'ANNPMT Pessimistic', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 15);

ROI\_title = uicontrol('Units', 'normalized', 'Position',[0.3 0.5+y\_prod\_offset 0.15/2 0.05], 'Style', 'text','Parent', TabHandles{finance\_page,1},...

'String', 'ROI%', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 15);

IRR\_title = uicontrol('Units', 'normalized', 'Position',[0.1 0.5+y\_prod\_offset 0.175/2 0.05], 'Style', 'text','Parent', TabHandles{finance\_page,1},...

'String', 'IRR%', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 15);

% Create the edit boxes to update

ALCC\_value = uicontrol('Units', 'normalized', 'Position',[0.3 0.8+y\_prod\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{finance\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

ANNPMT\_opt\_value = uicontrol('Units', 'normalized', 'Position',[0.3 0.725+y\_prod\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{finance\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

ANNPMT\_likely\_value = uicontrol('Units', 'normalized', 'Position',[0.3 0.65+y\_prod\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{finance\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

ANNPMT\_pess\_value = uicontrol('Units', 'normalized', 'Position',[0.3 0.575+y\_prod\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{finance\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

ROI\_value = uicontrol('Units', 'normalized', 'Position',[0.3+(0.15/2) 0.5+y\_prod\_offset 0.15/2 0.05], 'Style', 'edit','Parent', TabHandles{finance\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

IRR\_value = uicontrol('Units', 'normalized', 'Position',[0.1+0.175/2 0.5+y\_prod\_offset 0.175/2 0.05], 'Style', 'edit','Parent', TabHandles{finance\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

% Create the Labels for each value

savings\_title = uicontrol('Units', 'normalized', 'Position',[0.575 0.875+y\_prod\_offset 0.375 0.07], 'Style', 'text','Parent', TabHandles{finance\_page,1},...

'String', 'Expected Savings ($)', 'Visible', 'On','Backgroundcolor',[0.5 1 0], 'Foregroundcolor', 'black', 'FontSize', 20);

monthly\_saving\_title = uicontrol('Units', 'normalized', 'Position',[0.575 0.8+y\_prod\_offset 0.2 0.05], 'Style', 'text','Parent', TabHandles{finance\_page,1},...

'String', 'Monthly', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 15);

yearly\_saving\_title = uicontrol('Units', 'normalized', 'Position',[0.575 0.725+y\_prod\_offset 0.2 0.05], 'Style', 'text','Parent', TabHandles{finance\_page,1},...

'String', 'Yearly', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 15);

ten\_year\_saving\_title = uicontrol('Units', 'normalized', 'Position',[0.575 0.65+y\_prod\_offset 0.2 0.05], 'Style', 'text','Parent', TabHandles{finance\_page,1},...

'String', '10 Year Peiod', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 15);

twen\_year\_saving\_title = uicontrol('Units', 'normalized', 'Position',[0.575 0.575+y\_prod\_offset 0.2 0.05], 'Style', 'text','Parent', TabHandles{finance\_page,1},...

'String', '20 Year Period', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 15);

payback\_period\_title = uicontrol('Units', 'normalized', 'Position',[0.575 0.5+y\_prod\_offset 0.2/2 0.05], 'Style', 'text','Parent', TabHandles{finance\_page,1},...

'String', 'Payback (yr)', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 15);

NPV\_title = uicontrol('Units', 'normalized', 'Position',[0.8 0.5+y\_prod\_offset 0.15/2 0.05], 'Style', 'text','Parent', TabHandles{finance\_page,1},...

'String', 'NPV', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 15);

% Create the edit boxes to update

monthly\_saving\_value = uicontrol('Units', 'normalized', 'Position',[0.8 0.8+y\_prod\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{finance\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

yearly\_saving\_value = uicontrol('Units', 'normalized', 'Position',[0.8 0.725+y\_prod\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{finance\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

ten\_year\_saving\_value = uicontrol('Units', 'normalized', 'Position',[0.8 0.65+y\_prod\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{finance\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

twen\_year\_saving\_value = uicontrol('Units', 'normalized', 'Position',[0.8 0.575+y\_prod\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{finance\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

payback\_saving\_value = uicontrol('Units', 'normalized', 'Position',[0.575+0.1 0.5+y\_prod\_offset 0.1 0.05], 'Style', 'edit','Parent', TabHandles{finance\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

NPV\_value = uicontrol('Units', 'normalized', 'Position',[0.8+0.15/2 0.5+y\_prod\_offset 0.15/2 0.05], 'Style', 'edit','Parent', TabHandles{finance\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

% Create variables to be shared

persistent bar\_data\_finance

persistent yearly\_savings\_adj\_infla

% Main fucntion for finance

function finance\_graph()

% Create axis for graph

PlotOffset = 40;

haxes3 = axes('Parent', TabHandles{finance\_page,1}, ...

'Units', 'normalized', ...

'Position', [0.075 0.1 0.9 0.35]);

% Calcualte the yearly savings

yearly\_cost\_plot = kwhr\_avg\_found\*tariff\_rate\_normal\_found\*365;

yearly\_cost\_witsol\_plot = yearly\_cost\_plot - daily\_savings\*365;

% Creates a loop which effectively calculates compounding

% interest

for i = 1:1:20

if ((i == 10) & (battery\_installed == 1))

bar\_data\_finance(1,i) = (yearly\_cost\_witsol\_plot + cost\_battery\_input\*(1+inflation\_rate)^i); %no future value of money i ;

else

bar\_data\_finance(2,i) = yearly\_cost\_plot;

bar\_data\_finance(1,i) = yearly\_cost\_witsol\_plot;

end

% Inflation for normale rate and solar

yearly\_cost\_plot = yearly\_cost\_plot \*(1+inflation\_rate);

yearly\_cost\_witsol\_plot = yearly\_cost\_witsol\_plot \* (1+inflation\_rate);

end

% This is for if a negative value (i.e savings) are found then

% the bar chart cannot plot and this is swaps the axis and type

% of bar graph

if (yearly\_cost\_witsol\_plot > 0)

buddy = bar(haxes3,bar\_data\_finance','stacked','FaceColor',[0 .8 .5],'EdgeColor','yellow','LineWidth',1.5);

else

buddy = bar(haxes3,bar\_data\_finance','FaceColor',[0 .8 .5],'EdgeColor','yellow','LineWidth',1.5);

end

buddy(2).FaceColor = [.9 .1 .1];

% Label, Dimension and Legent the GRPAH

title('Yearly Cost of Electricity Comparison Between Normal Vs. Solar/Battery','Color','yellow','FontSize', 16);

xlabel('Years','FontSize', 16); ylabel('Cost ($AUD)','FontSize', 16);

xlim([0, 21]);

legend({'Solar/Battery','Normal'},'Location','northwest');

set(gca, ...

'Box' , 'off' , ...

'TickDir' , 'out' , ...

'TickLength' , [.01 .01] , ...

'XMinorTick' , 'on' , ...

'YMinorTick' , 'on' , ...

'YGrid' , 'off' , ...

'XGrid' , 'on' , ...

'Color',Grey, 'FontSize', 13,...

'XColor' , 'yellow', ...

'YColor' , 'yellow', ...

'LineWidth' , 2 );

% End of Graph Labelling

% This was another graph used on another tab to use in report

haxes7 = axes('Parent', TabHandles{7,1}, ...

'Units', 'normalized', ...

'Position', [0.08 0.15 0.9 0.75]);

% Added in for when you make money with solar

if (yearly\_cost\_witsol\_plot > 0)

buddy7 = bar(haxes7,bar\_data\_finance','stacked','FaceColor',[0 .8 .5],'EdgeColor','yellow','LineWidth',1.5);

else

buddy7 = bar(haxes7,bar\_data\_finance','FaceColor',[0 .8 .5],'EdgeColor','yellow','LineWidth',1.5);

end

buddy7(2).FaceColor = [.9 .1 .1];

% Label, Dimension and Legent the GRPAH

title('Yearly Cost of Electricity Comparison Between Normal Vs. Solar/Battery','Color','black','FontSize', 16);

xlabel('Years','FontSize', 16); ylabel('Cost ($AUD)','FontSize', 16);

xlim([0, 21]);

legend({'Solar/Battery','Normal'},'Location','northwest');

set(gca, ...

'Box' , 'off' , ...

'TickDir' , 'out' , ...

'TickLength' , [.01 .01] , ...

'XMinorTick' , 'on' , ...

'YMinorTick' , 'on' , ...

'YGrid' , 'off' , ...

'XGrid' , 'on' , ...

'Color','white', 'FontSize', 13,...

'XColor' , 'black', ...

'YColor' , 'black', ...

'LineWidth' , 2 );

% End of Graph Labelling

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Define Tab 5 content (TAB JUMP)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Create the UI Handles for the labels and Values on Display Page

% Create tab reference for parent handles

y\_disp\_offset = 0.025;

% Set up page number for referencing

display\_page = 5;

% Create the Labels for each value

disp\_prod\_title = uicontrol('Units', 'normalized', 'Position',[0.375 0.5+y\_disp\_offset 0.15 0.05], 'Style', 'text','Parent', TabHandles{display\_page,1},...

'String', 'Daily Production', 'Visible', 'On','Backgroundcolor', 'green', 'Foregroundcolor', 'black', 'FontSize', 15);

% Edit boxes for production

disp\_prod\_value = uicontrol('Units', 'normalized', 'Position',[0.375 0.43+y\_prod\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{display\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

% Create the static text for production

disp\_stored\_title = uicontrol('Units', 'normalized', 'Position',[0.375 0.195+y\_disp\_offset 0.15 0.05], 'Style', 'text','Parent', TabHandles{display\_page,1},...

'String', 'Daily Storage', 'Visible', 'On','Backgroundcolor', 'yellow', 'Foregroundcolor', 'black', 'FontSize', 15);

% Edit boxes for production

disp\_stored\_value = uicontrol('Units', 'normalized', 'Position',[0.375 0.125+y\_prod\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{display\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

% Create the static text for production

disp\_used\_title = uicontrol('Units', 'normalized', 'Position',[0.725 0.195+y\_disp\_offset 0.15 0.05], 'Style', 'text','Parent', TabHandles{display\_page,1},...

'String', 'Daily Usage', 'Visible', 'On','Backgroundcolor', 'red', 'Foregroundcolor', 'black', 'FontSize', 15);

% Edit boxes for production

disp\_used\_value = uicontrol('Units', 'normalized', 'Position',[0.725 0.125+y\_prod\_offset 0.15 0.05], 'Style', 'edit','Parent', TabHandles{display\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

% Create the static text for production

disp\_exported\_title = uicontrol('Units', 'normalized', 'Position',[0.725 0.5+y\_disp\_offset 0.07 0.05], 'Style', 'text','Parent', TabHandles{display\_page,1},...

'String', 'Export', 'Visible', 'On','Backgroundcolor', 'green', 'Foregroundcolor', 'black', 'FontSize', 15);

% Edit boxes for production

disp\_exported\_value = uicontrol('Units', 'normalized', 'Position',[0.8 0.5+y\_prod\_offset 0.07 0.05], 'Style', 'edit','Parent', TabHandles{display\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

% Create the static text for production

disp\_imported\_title = uicontrol('Units', 'normalized', 'Position',[0.725 0.43+y\_disp\_offset 0.07 0.05], 'Style', 'text','Parent', TabHandles{display\_page,1},...

'String', 'Import', 'Visible', 'On','Backgroundcolor', 'red', 'Foregroundcolor', 'black', 'FontSize', 15);

% Edit boxes for production

disp\_imported\_value = uicontrol('Units', 'normalized', 'Position',[0.8 0.43+y\_prod\_offset 0.07 0.05], 'Style', 'edit','Parent', TabHandles{display\_page,1},...

'String', '-', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

% Create units box

disp\_imported\_title = uicontrol('Units', 'normalized', 'Position',[0. 0.96 0.07 0.04], 'Style', 'text','Parent', TabHandles{display\_page,1},...

'String', 'Units - kWhr', 'Visible', 'On','Backgroundcolor', Grey, 'Foregroundcolor', 'black', 'FontSize', 10);

%% Save the TabHandles in guidata

% Used to save the tabs

guidata(hTabFig,TabHandles);

%% Make Tab 1 active

% Makes a selected tab active after the initialisation

TabSellectCallback(0,0,1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Background Images (TAB JUMP)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Create the background pictures for each tab

% This would not work in a loop for some reason so it was duplicated here

% in the code

%%%%%%%%%%%%%%%%%1

% (1)Create axis which covers the entire GUI workspace

background\_picture = axes('Parent', TabHandles{1,1},'unit', 'pixels', 'position', [1,1,MaxWindowX,MaxWindowY]);

% (2)import the background image and show it on the axes

background\_image = imread('homepage\_solar\_background.jpg'); imagesc(background\_image);

% (3) Turn the axis off and stop plotting from being permitable over the background

set(background\_picture,'handlevisibility','off','visible','off')

% (4)Ensure all the other objects in the GUI are infront of the background

uistack(background\_picture, 'bottom');

%%%%%%%%%%%%%%%%%2

% (1)Create axis which covers the entire GUI workspace

background\_picture = axes('Parent', TabHandles{2,1},'unit', 'pixels', 'position', [1,1,MaxWindowX,MaxWindowY]);

% (2)import the background image and show it on the axes

background\_image = imread('homepage\_solar\_background.jpg'); imagesc(background\_image);

% (3) Turn the axis off and stop plotting from being permitable over the background

set(background\_picture,'handlevisibility','off','visible','off')

% (4)Ensure all the other objects in the GUI are infront of the background

uistack(background\_picture, 'bottom');

%%%%%%%%%%%%%%%%%3

% (1)Create axis which covers the entire GUI workspace

background\_picture = axes('Parent', TabHandles{3,1},'unit', 'pixels', 'position', [1,1,MaxWindowX,MaxWindowY]);

% (2)import the background image and show it on the axes

background\_image = imread('homepage\_solar\_background.jpg'); imagesc(background\_image);

% (3) Turn the axis off and stop plotting from being permitable over the background

set(background\_picture,'handlevisibility','off','visible','off')

% (4)Ensure all the other objects in the GUI are infront of the background

uistack(background\_picture, 'bottom');

%%%%%%%%%%%%%%%%%4

% (1)Create axis which covers the entire GUI workspace

background\_picture = axes('Parent', TabHandles{4,1},'unit', 'pixels', 'position', [1,1,MaxWindowX,MaxWindowY]);

% (2)import the background image and show it on the axes

background\_image = imread('homepage\_solar\_background.jpg'); imagesc(background\_image);

% (3) Turn the axis off and stop plotting from being permitable over the background

set(background\_picture,'handlevisibility','off','visible','off')

% (4)Ensure all the other objects in the GUI are infront of the background

uistack(background\_picture, 'bottom');

%%%%%%%%%%%%%%%%%%5

% (1)Create axis which covers the entire GUI workspace

background\_picture = axes('Parent', TabHandles{5,1},'unit', 'pixels', 'position', [10,10,MaxWindowX-10,MaxWindowY-10]);

% (2)import the background image and show it on the axes

background\_image = imread('diagram\_of\_system.png'); imagesc(background\_image);

% (3) Turn the axis off and stop plotting from being permitable over the background

set(background\_picture,'handlevisibility','off','visible','off')

% (4)Ensure all the other objects in the GUI are infront of the background

uistack(background\_picture, 'bottom');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Cost Analysis (TAB JUMP)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% This sections computes all the costing for the GUI

% Create persistent Variables to be used

persistent inflation\_rate

persistent investment\_cost

persistent IRR\_val

persistent NPV

% Main function for cost analysis

function cost\_analysis(daily\_savings)

% Creates the financial values

inflation\_rate = 0.03; mortgage\_rates = [0.01 0.035 0.06];

discount\_rate = 0.04;

n\_years = 20; % or maybe 25

x\_unitless = (1 + inflation\_rate)/(1 + discount\_rate);

pa = (1 - x\_unitless^(n\_years))/(1 - x\_unitless);

pa1 = x\_unitless\*pa ;

% These were not not used

% maintenance\_cost = 0.015\*cost\_solar\_input\*20

% salvage\_cost = solar\_size\_input\*0.21\*1000;

% LCC Analysis

LCC = cost\_solar\_input + cost\_battery\_input\*2 ;

% ALCC Analysis

ALCC = LCC/pa;

for i = 1:1:3

mort = mortgage\_rates(1,i);

ANNPMT(1,i) = LCC \*mort\*( ((1+mort)^n\_years) / (((1+mort)^n\_years)-1) ) ;

electricity\_cost\_ANNPMT(1,i) = ANNPMT(1,i)/(kw\_produced\_daily\*365);

end

% Calculates COE

electricity\_cost\_ALCC = ALCC/(kw\_produced\_daily\*365);

investment\_cost = cost\_solar\_input + cost\_battery\_input\*2;

payback\_period = investment\_cost/ (daily\_savings\*365);

% Investment\_cost and IRR Calculation

yearly\_savings\_adj\_infla = daily\_savings\*365;

IRR\_sym = sym('IRR\_sym') ;

IRR\_eqn = 0;

for i = 1:1:20

if i == 10

cash\_flow\_in(1,i) = yearly\_savings\_adj\_infla - cost\_battery\_input ;

else

cash\_flow\_in(1,i) = yearly\_savings\_adj\_infla ;

end

yearly\_savings\_adj\_infla = yearly\_savings\_adj\_infla \* (1+inflation\_rate);

cash\_flows\_discounted(1,i) = cash\_flow\_in(1,i) / ((1 + discount\_rate));

IRR\_eqn = IRR\_eqn + ( cash\_flows\_discounted(1,i) / (1 + IRR\_sym)^i );

end

% NPV Calculation and ROI

NPV = sum(cash\_flows\_discounted)-investment\_cost ;

% I think the method is a better indication not the blanked

% out version (this takes into account inflation and discount)

% ROI = ((daily\_savings\*365\*20 - investment\_cost) / investment\_cost)\*100/20

ROI = ((NPV - investment\_cost )\*100) / investment\_cost;

% Solve IRR Equation

IRR\_eqn = IRR\_eqn - investment\_cost == 0;

IRR\_sol = real(double(solve(IRR\_eqn, IRR\_sym)));

% Find the positive value

Positive = IRR\_sol > 0;

IRR\_sol(~Positive) = 0;

IRR\_val = 0;

element = 0;

if (IRR\_sol(1,1) == 0)

for i = 1:1:4

element = IRR\_sol(i,1);

if element > IRR\_val

IRR\_val = element;

end

end

end

%Display the IRR Value

% disp('IRR Value:')

% disp(IRR\_val\*100)

% Updates all the values to the GUI

Update\_Values\_cost(electricity\_cost\_ALCC,electricity\_cost\_ANNPMT,daily\_savings,payback\_period,ROI)

end

% Used to update the production values

function Update\_Values\_prod(kwhr\_avg\_found,kw\_produced\_daily,daily\_storage,tariff\_rate\_normal\_found,daily\_exported,daily\_imported,daily\_savings)

% Sends to the economic tab the calculate values

set(daily\_usuage\_value,'string', num2str(kwhr\_avg\_found)); disp(kwhr\_avg\_found)

set(daily\_production\_value,'string', num2str(kw\_produced\_daily)); disp(kw\_produced\_daily)

set(daily\_storage\_value , 'String', num2str(daily\_storage)); disp(daily\_storage)

set(daily\_exported\_value , 'String', num2str(daily\_exported)) ; disp(daily\_exported)

set(daily\_normal\_cost\_value , 'String', num2str(kwhr\_avg\_found\*tariff\_rate\_normal\_found)) ; disp(kwhr\_avg\_found\*tariff\_rate\_normal\_found)

set(daily\_import\_cost\_value , 'String', num2str(daily\_imported\*tariff\_rate\_normal\_found)); disp(daily\_imported\*tariff\_rate\_normal\_found)

set(daily\_export\_cost\_value , 'String', num2str(daily\_exported\*solar\_rate\_feedin\_found)) ; disp(daily\_exported\*solar\_rate\_feedin\_found)

set(daily\_savings\_cost\_value , 'String', num2str(daily\_savings)) ; disp(daily\_savings)

set(disp\_used\_value,'string', num2str(kwhr\_avg\_found)); disp(source\_energy)

set(disp\_prod\_value,'string', num2str(kw\_produced\_daily));

set(disp\_stored\_value , 'String', num2str(daily\_storage));

set(disp\_exported\_value , 'String', num2str(daily\_exported));

set(disp\_imported\_value , 'String', num2str(daily\_imported));

end

% Used to update the cost values

function Update\_Values\_cost(electricity\_cost\_ALCC,electricity\_cost\_ANNPMT,daily\_savings,payback\_period,ROI)

% Sends to the economic tab the calculate values

set(ALCC\_value,'string', num2str(electricity\_cost\_ALCC)); disp(electricity\_cost\_ALCC)

set(ANNPMT\_opt\_value,'string', num2str(electricity\_cost\_ANNPMT(1,1))); disp(electricity\_cost\_ANNPMT(1,1))

set(ANNPMT\_likely\_value , 'String', num2str(electricity\_cost\_ANNPMT(1,2))); disp(electricity\_cost\_ANNPMT(1,2))

set(ANNPMT\_pess\_value , 'String', num2str(electricity\_cost\_ANNPMT(1,3))); disp(electricity\_cost\_ANNPMT(1,3))

set(IRR\_value, 'String', num2str(IRR\_val\*100)); disp(IRR\_val\*100)

set(ROI\_value, 'String', num2str(ROI)); disp(ROI)

set(monthly\_saving\_value,'string', num2str(daily\_savings\*30)); disp(daily\_savings\*30)

set(yearly\_saving\_value,'string', num2str(daily\_savings\*365)); disp(daily\_savings\*365)

set(ten\_year\_saving\_value,'string', num2str(daily\_savings\*3650)); disp(daily\_savings\*3650)

set(twen\_year\_saving\_value,'string', num2str(daily\_savings\*3650\*2)); disp(daily\_savings\*3650\*2)

set(payback\_saving\_value,'string', num2str(payback\_period)); disp(payback\_period)

set(NPV\_value, 'String', num2str(NPV)); disp(NPV)

end

% Used to indicate to the user the time

function clock\_disp()

% Retrives the time from the computer

time\_all = fix(clock);

hr = time\_all(1,4);

min = time\_all(1,5);

% Creates the axis for plotting

uicontrol('Style','text','Units', 'normalized','Position',[0.0 0.0 0.04 0.03],...

'string',sprintf('%d:%d',hr,min),'BackgroundColor', White,...

'HorizontalAlignment','left','FontName','arial','FontWeight','bold', ...

'FontSize',14);

axis off;

end

% Used to indicate to the user what percentage is completed

function progress\_bar(progress\_step)

% Creates the axis for plotting

axes('Units','normalized','Position',[0.3 0.05 0.4 0.07],'Parent', TabHandles{1,1});

axis off;

% Fills the shaded area

area1 = area([0 progress\_step/number\_of\_inputs],[1 1]);

set(gca,'xtick',[]); set(gca,'ytick',[]);

set(gca,'xlim',[0 1]);

set(gca,'ylim',[0 1]);

set(area1,'FaceColor','yellow');

% Displays the text percentage numerically

uicontrol('Style', 'text','Units', 'normalized','Position', [0.3 0.05-0.038 0.4 0.038],'Parent', TabHandles{1,1},...

'string',sprintf('Progress: %i%%',round(progress\_step\*100/number\_of\_inputs)),'BackgroundColor',Grey,...

'HorizontalAlignment','center','FontName','arial','FontWeight','bold', ...

'FontSize', 12);

end

% Used in the rest of the program

function reset(hObject, eventdata)

% Resets all the inputs and displays

set(gas\_main\_value, 'String', '-'); set(pool\_value, 'String', '-');

set(solar\_size\_value, 'String', '-'); set(solar\_cost\_value, 'String', '-');

set(battery\_size\_value, 'String', '-' ); set(battery\_cost\_value, 'String', '-' );

set(tilt\_value, 'String', '-' ); set(orientation\_value, 'String', '-');

set(bill\_value, 'String', '-' ); set(occupants\_value, 'String', '-');

set(state\_value, 'String', '-') ; set(postal\_value, 'String', '-');

set(tariff\_value, 'String', '-'); set(supplier\_value, 'String', '-') ;

% pauses, closes, displays error and restarts the GUI

pause(0.75)

close(gcbf)%

errordlg('Error Restarting, Reboot Initiated','Setup Error')

The\_Solar\_Solution\_GUI()

set(prefill\_button,'Visible','On')

end

% Used in Prefill function to clear all uipanels

function all\_visible\_off(hObject, eventdata)

% Turns off all the button when prefill is selected

set(gas\_mains\_question,'Visible','Off')

set(button\_yes\_gas\_mains,'Visible','Off')

set(button\_no\_gas\_mains,'Visible','Off')

set(pool\_question,'Visible','Off')

set(button\_yes\_pool,'Visible','Off')

set(button\_no\_pool,'Visible','Off')

set(percentage\_question,'Visible','Off')

set(percentage\_popupmenu,'Visible','Off')

set(percentage\_next\_button,'Visible','Off')

set(performance\_question,'Visible','Off')

set(performance\_popupmenu,'Visible','Off')

set(performance\_next\_button,'Visible','Off')

set(text\_solar\_question,'Visible','OFF')

set(button\_yes\_solar,'Visible','OFF')

set(button\_no\_solar,'Visible','OFF')

set(solar\_size\_next\_button,'Visible','off')

set(KW\_popupmenu,'Visible','off')

set(solar\_size\_question,'Visible','off')

set(text\_solar\_question,'Visible','OFF')

set(button\_yes\_solar,'Visible','OFF')

set(button\_no\_solar,'Visible','OFF')

set(cost\_solar\_question,'Visible','OFF')

set(cost\_popupmenu,'Visible','OFF')

set(cost\_next\_button,'Visible','OFF')

set(text\_battery\_question,'Visible','OFF')

set(button\_yes\_battery,'Visible','OFF')

set(button\_no\_battery,'Visible','OFF')

set(battery\_size\_question,'Visible','OFF')

set(KWHR\_popupmenu,'Visible','OFF')

set(battery\_size\_next\_button,'Visible','OFF')

set(text\_battery\_question,'Visible','OFF')

set(button\_yes\_battery,'Visible','OFF')

set(button\_no\_battery,'Visible','OFF')

set(cost\_battery\_question,'Visible','Off')

set(battery\_cost\_popupmenu,'Visible','Off')

set(battery\_cost\_next\_button,'Visible','Off')

set(text\_roof\_question,'Visible','OFF')

set(tilt\_popupmenu,'Visible','OFF')

set(roof\_next\_button,'Visible','OFF')

set(orientation\_next\_button,'Visible','Off')

set(orientation\_edit\_display,'Visible','Off')

set(text\_orientation\_question,'Visible','Off')

set(compass\_image,'Visible','Off')

set(radio\_north\_button,'Visible','Off');

set(radio\_north\_west\_button,'Visible','Off')

set(radio\_east\_button,'Visible','Off');

set(radio\_north\_east\_button,'Visible','Off')

set(radio\_south\_button,'Visible','Off');

set(radio\_south\_east\_button,'Visible','Off')

set(radio\_south\_west\_button,'Visible','Off')

set(state\_display\_button,'Visible','Off')

set(state\_next\_button,'Visible','Off')

set(text\_state\_question,'Visible','Off')

set(postcode\_edit,'Visible','Off')

set(text\_state\_codes,'Visible','Off');

set(post\_code\_table,'Visible','Off')

set(text\_bill\_question,'Visible','Off')

set(bill\_edit,'Visible','Off')

set(bill\_next\_button,'Visible','Off')

set(bill\_skip\_button,'Visible','Off')

set(number\_people\_question,'Visible','Off')

set(people\_popupmenu,'Visible','Off')

set(people\_next\_button,'Visible','Off')

end

% Used in prefilling the GUI with data instead of the prompt process

function prefill(hObject, eventdata)

% Progress bar and turn off uihandles

progress\_bar(number\_of\_inputs);

all\_visible\_off()

loading\_bar = waitbar(0,'Please wait...');

steps = 1000;

for step = 1:steps

% computations take place here

waitbar(step / steps)

end

% Set values to be prefiled here...

gas\_mains\_input = 0;

pool\_input = 0;

percentage\_input = 30;

performance\_input = 0.85;

solar\_installed = 1;

solar\_size\_input = 5;

cost\_solar\_input = 7000;

battery\_installed = 1;

battery\_size\_input = 10;

cost\_battery\_input = 9700;

orientation\_input = 1;

bill\_input = 0;

number\_people\_input = 3;

%%% Change state

state\_input = 7000;

%%% Change Roof tilt

roof\_tilt\_input = 12;

% Set values on the GUI for display

set(pool\_value, 'String', 'No')

set(solar\_size\_value, 'String', num2str(solar\_size\_input))

set(solar\_cost\_value, 'String', num2str(cost\_solar\_input))

set(battery\_size\_value, 'String', num2str(battery\_size\_input) )

set(battery\_cost\_value, 'String', num2str(cost\_battery\_input) )

set(orientation\_value, 'String', 'North')

set(bill\_value, 'String', num2str(bill\_input) )

set(occupants\_value, 'String', num2str(number\_people\_input))

set(gas\_main\_value, 'String', 'No')

set(tilt\_value, 'String', num2str(roof\_tilt\_input) )

set(state\_value, 'String', 'QLD') ;

set(postal\_value, 'String', '4814')

set(tariff\_value, 'String', '11')

set(supplier\_value, 'String', 'Ergon')

set(enter\_gui\_button,'Visible','On')

close(loading\_bar)

% Calls the main function for calculations

PSH\_and\_KW\_Calc(solar\_size\_input,performance\_input, roof\_tilt\_input,state\_input...

,number\_people\_input,gas\_mains\_input, pool\_input, battery\_size\_input, solar\_installed, battery\_installed)

% Runs through each tab giving the user an overview of the

% program

TabSellectCallback(0,0,8);

pause(0.75)

TabSellectCallback(0,0,7);

pause(0.75)

TabSellectCallback(0,0,6);

pause(0.75)

TabSellectCallback(0,0,5);

pause(0.75)

TabSellectCallback(0,0,4);

pause(0.75)

TabSellectCallback(0,0,3);

pause(0.75)

TabSellectCallback(0,0,2);

progress\_bar(0);

set(enter\_gui\_button,'Visible','On')

end

% End of Main GUI Loop

end

%% Callback for Tab Selection (TAB JUMP)

function TabSellectCallback(~,~,SelectedTab)

%{

All tab selection pushbuttons are greyed out and uipanels are set to visible off,

then the selected panel is made visible and it's selection pushbutton is highlighted

%}

% Set up some varables

TabHandles = guidata(gcf);

NumberOfTabs = size(TabHandles,1)-2;

White = TabHandles{NumberOfTabs+2,2}; % White

BGColor = TabHandles{NumberOfTabs+2,3}; % Light Grey

% Turn all tabs off

for TabCount = 1:NumberOfTabs

set(TabHandles{TabCount,1}, 'Visible', 'off');

set(TabHandles{TabCount,2}, 'BackgroundColor', BGColor);

end

% Enable the selected tab

set(TabHandles{SelectedTab,1}, 'Visible', 'on');

set(TabHandles{SelectedTab,2}, 'BackgroundColor', White);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%% End of Script for Solar Solution GUI Display %%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%