Technical University of Cluj-Napoca

MATLAB Project – MOVIE TRACKING
Theme: an interactive movie traking and analysis system with genre-specific movie selection

Student:
Pedolu Andreea-Emilia

Coordinator:
Cîrlugea Mihaela
Farago Paul

2024-2025

TABLE OF CONTENTS

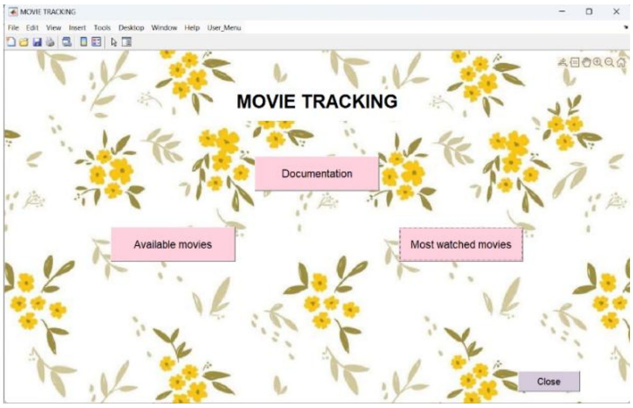
GENERAL FUNCTIONALITY

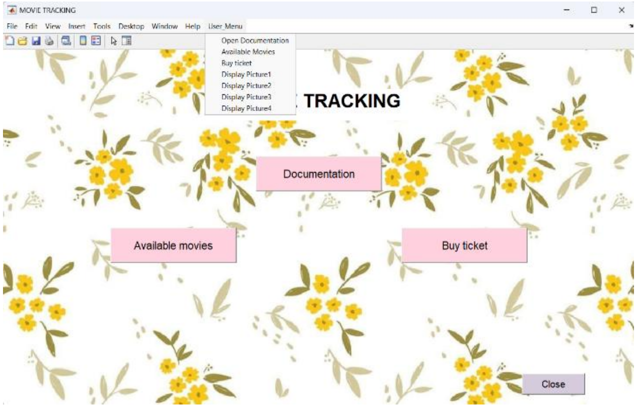
The project is composed of a menu in which there are three important buttons, one for the documentation, one for “Available movies” and one “Buy ticket” and a small close button.

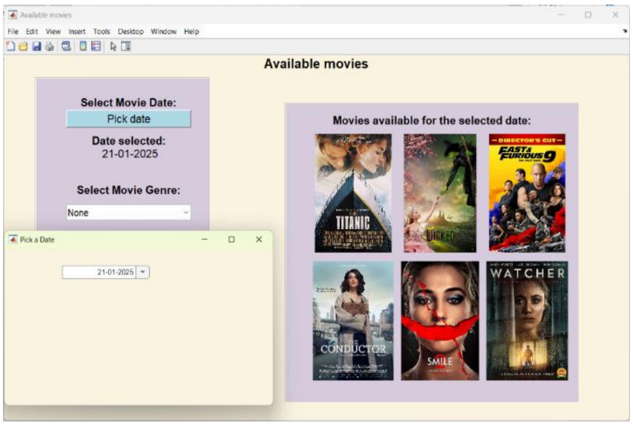
The Available movies button sends the user to another page in which after a date is selected from the date picker it is displayed on the screen and the movies for the selected date are displayed or a message box will be displayed in which the user will be informed in the selected date there are no movies available. After this action, the drop down menu will be activated and the user will be able to select a specific movie genre. Once a genre is selected and another date is picked, there will be displayed only the movies for the selected genre and movie or a messege box will appear informing the user there are no movies for the selected day or genre.

The Buy ticket button sends the user to a page where the user can buy one or more tickets for movies and see the most watched movies from last week. It is an interactive interface in which the person adjust the quantity of tickets, enters the date of the movie (which is found in the Available movies section) and selects the movie. In the same time, the user can see the most watched movies from last week. After it’s selected a day of the week, some movies and the number of views for every movie the user can press the button “Show Most Watched Movies” and a message box will appear with the specific number for each day of the week and even if the user didn’t select a day of the week. If the button “Plot most watched movies” is pressed then a bar charted plot will appear with specific number of views.

The project contains as well a user menu in which there are available the main important buttons “Open Documentation”, “Available movies”, “Buy ticket” but there are also some pictures.



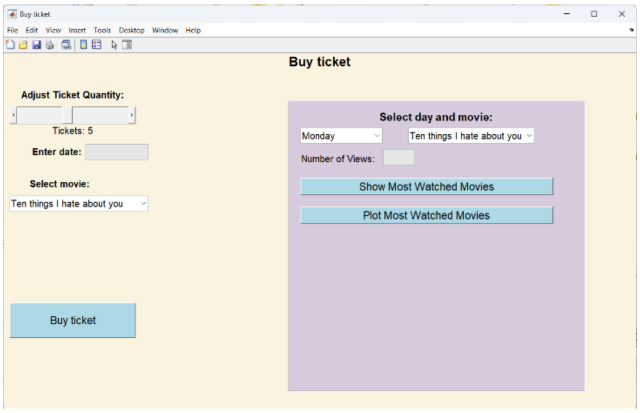




Date selected:
21-01-2025

None

21-01-2025



Tickets: 5

Ten things I hate about you

Monday

Ten things I hate about you

INTRODUCTION

History of matlab

MATLAB, short for "MATrix LABoratory," is a proprietary multi-paradigm programming language and numeric computing environment developed by MathWorks. Initially conceived in the 1960s by mathematician Cleve Moler, MATLAB evolved from a basic matrix calculator to a powerful platform widely used for numeric computing, algorithm implementation, data visualization, user interface creation, and integration with other programming languages. It incorporates an optional symbolic computing toolbox powered by the MuPAD engine and a specialized package, Simulink, which facilitates graphical multi-domain simulation and model-based design for dynamic systems.

The origins of MATLAB trace back to Moler's work during his PhD thesis and his tenure as a math professor at the University of New Mexico, where he developed it for teaching purposes. In its early stages, MATLAB was a simple interactive tool with 71 built-in functions, shared freely among universities in the late 1970s. Its popularity grew as Moler distributed copies to academic institutions, fostering a strong following within university mathematics departments.

The 1980s marked a turning point when Moler partnered with John N. Little and Steve Bangert to reprogram MATLAB in C, creating a formal programming language and introducing toolboxes for specialized mathematical tasks. This effort culminated in MATLAB's first commercial release at the 1984 Automatic Control Conference in Las Vegas, supported by the founding of MathWorks, Inc. The initial commercial adoption began with a sale to MIT in 1985, and by the decade's end, MATLAB had gained traction across academia and industry. Much of its expansion was driven by Stanford students who used MATLAB in academic settings and later introduced it to private sector applications.

Over time, MATLAB adapted to emerging technologies and platforms, transitioning to operating systems like Unix, VAX, and Sun Microsystems. The 1987 release of Version 3 marked significant advancements, and the 1990s saw the introduction of the MATLAB compiler, enhancing its versatility. Major updates included replacing original LINPACK and EISPACK routines with a Fortran-based library in 2000 and the launch of the Parallel Computing Toolbox in 2004, later augmented with GPU support in 2010. The software's extensive toolbox ecosystem, built in collaboration with field experts, played a crucial role in its widespread adoption across engineering, science, and economics.

Today, MATLAB serves more than four million users worldwide, spanning diverse fields, and is employed by over 5,000 universities for education and research. Its open-source alternatives, such as GNU Octave and Scilab, offer compatibility and similar functionality, but MATLAB remains a cornerstone in the realm of computational software, bridging academia and industry through its robust capabilities.

ELEMENTS CONTAINED IN THE PROJECT

The date picker is an interesting feature that helps the user to select a date and it will be displayed on the screen and different movies will appear based on the selected date.

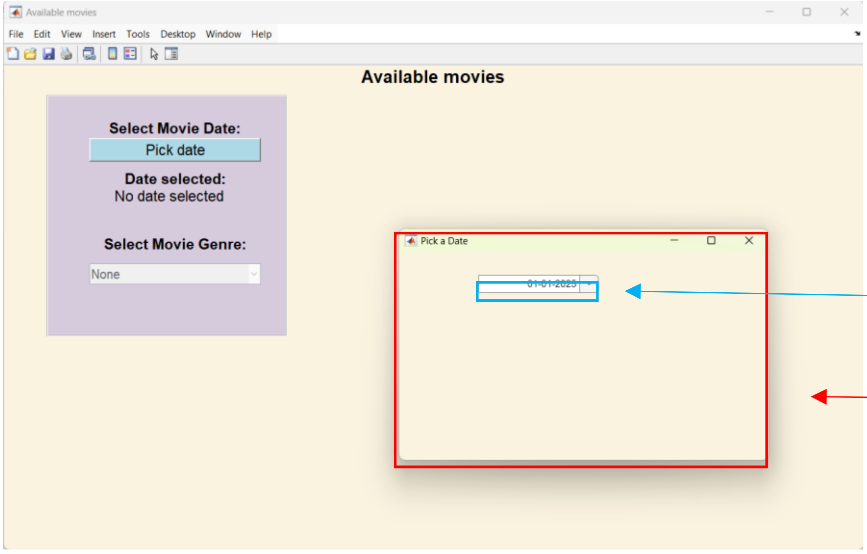
The date picker is called with the button “Pick a date” in the function available_movies(fig)

```
%button to pick a date
   uicontrol('Style', 'pushbutton', ...
           'Units', 'normalized', ...
           'Fontsize', 14, ...
           'BackgroundColor', '#ADD8E6', ...
           'Position', [0.1 0.8 0.2 0.05], ...
           'String', 'Pick date', ...
           'Callback', @(src, event) pick_date_callback(src, event, fig));
```

The callback is an anonymous function which is in the function file pick_date_callback.m
The expression 'Callback', @(src, event) is specifying the callback function for a UI control in MATLAB, using an anonymous function.

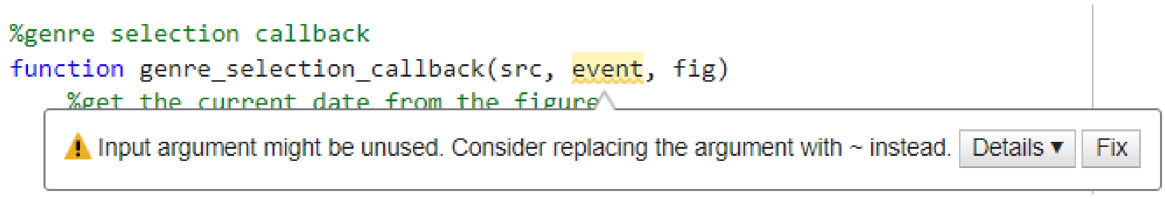
```
function pick_date_callback(~, ~, fig)
   %create a new figure for the date picker
   date_picker_fig = uifigure('Name', 'Pick a Date', ...
                     'NumberTitle', 'off', ...
                     'Color', '#FAF3E0', ...
                     'Units', 'normalized', ...
                     'Position', [0.1 0.13 0.3 0.3]);

   %add a date picker widget to the new figure
   uidatepicker(date_picker_fig, 'DisplayFormat', 'dd-MM-yyyy', ...
          'Position', [100 210 150 22], ...
          'Value', datetime(2025, 1, 1), ...
          'ValueChangedFcn', @(dp, event) date_change(dp, event, fig));
end
```



Based on my research the date picker can be implemented without anonymous function but it is easier this way. An anonymous function in MATLAB is a one-line, unnamed function that is defined using the @ syntax. It is often used for short, simple functions or as a way to pass additional arguments to callback functions.

In anonymous functions if one of the parameters of the function isn't used in the callback then it is replaced with "~". For example, I have a warning from matlab that suggest me to do this in the next callback.



```
%genre selection callback
function genre_selection_callback(src, event, fig)
    %get the current date from the figure
```

⚠ Input argument might be unused. Consider replacing the argument with ~ instead.

In my case, the function pick_date_callback which is the callback of the pick a date button creates a new figure and in it is added the date picker.

In the date picker is another anonymous function date_change(dp, event, fig)

```
%callback function for the date picker
function date_change(dp, ~, fig)
   selected_date = datestr(dp.Value, 'dd-mm-yyyy'); % Get the selected date
   disp(['Date selected: ', selected_date]);

   %update the text control with the selected date
   date_display = fig.UserData.date_display;
   date_display.String = [' ', selected_date];

   %enable the genre selection dropdown after a date is selected
   genre_dropdown = fig.UserData.genre_dropdown;
   genre_dropdown.Enable = 'on';

   %call the display_movies function to handle movie display
   display_movies(fig, selected_date);
end
```

Here datestr converts the datetime (dp.Value) into a readable string andis saved in the variable selected_date, afterwards the selected is displayed in the command window.

Afterwards the selected date is displayed.

In the function available_movies(fig) is originally created the text for the date display

```
%"Date selected:"
   uicontrol('Style', 'text', ...
          'Units', 'normalized', ...
          'FontSize', 14, ...
          'FontWeight', 'bold', ...
          'BackgroundColor', '#D6CADD', ...
          'Position', [0.1 0.735 0.2 0.05], ...
          'String', 'Date selected:');

   % No date selected
   date_display = uicontrol('Style', 'text', ...
                    'Units', 'normalized', ...
                    'FontSize', 14, ...
                    'FontWeight', 'normal', ...
                    'BackgroundColor', '#D6CADD', ...
                    'Position', [0.1 0.7 0.2 0.05], ...
                    'String', 'No date selected   ');

   % Store date_display handle in the figure's UserData
   fig.UserData.date_display = date_display;
```

Here the date_display is saved in the UserData.date_display of the figure fig and later is accesed in the function date_change(dp, ~, fig)

```
date_display = fig.UserData.date_display;
```

To display the selected day on the screem I used the String property which updates the visible text and the [' ', selected_date] creates a string by concatenating a space with the value of the selected_date for better visibility.

After the display movies function is called to display the movies.

Another interesting element is the genre drop down menu.
It is defined in the function available_movies(fig).

```
%Dropdown menu
    genre_dropdown = uicontrol('Style', 'popupmenu', ...
                       'Units', 'normalized', ...
                       'Position', [0.1, 0.55, 0.2, 0.04], ...
                       'FontSize', 12, ...
                       'BackgroundColor', '#FFFFFF', ...
                       'String', {'None', 'Animation', 'Action', 'Drama', 'Comedy', 'Horror',
'Romance', 'Mystery', 'Fantasy'}, ...
                       'Enable', 'off', ...
                       'Callback', @(src, event) genre_selection_callback(src, event, fig));

    % Store genre_dropdown handle in the figure's UserData
    fig.UserData.genre_dropdown = genre_dropdown;
```

It is initially off : 'Enable', 'off', ... and it has as callback the genre_selection_callback(src, event, fig)). The genre_dropdown is saved in the UserData.date_display of the figure fig and later is accesed in the function date_change(dp, ~, fig) where after a date is picked the genre drop down is on and the user can select a genre.

```
% Enable the genre selection dropdown after a date is selected
   genre_dropdown = fig.UserData.genre_dropdown;
   genre_dropdown.Enable = 'on';
```

Now about the genre_selection_callback(src, event, fig)

```
function genre_selection_callback(src, event, fig)
   %get the current date from the figure
   date_display = fig.UserData.date_display;
   selected_date = date_display.String;  %get the date string

   %if a date is selected, get the selected genre
   genres = src.String;
   selected_option = genres{src.Value};  %get the selected genre

   %call the display_movies function
   display_movies(fig, selected_date);
end
```

Here I firstly acces the reference of the displayed date on the screen from fig.UserData.date_display so that date_display variable now holds the handle of the displayed date.

After with the String property its accessed the text currently displayed and the value of date_display.String is then assigned to the variable selected_date.

I get the list of all genre options that are displayed in the dropdown with src.String because src refers to the handle of the dropdown menu and .String in this case contains the list of all the options available in the dropdown.

With the genres{src.Value} it's **accessed the selected option** from the genres array, because src.Value is providing the index of the the currently selected option in the dropdown list.
If the user has selected the first option ('None'), src.Value will be 1.

And after the function display_movies(fig, selected_date) is called.

About the function display_movies(fig, selected_date)

```
function display_movies(fig, selected_date)

  group=uibuttongroup('Visible','on',...
              'BackgroundColor','#D6CADD',...
              'ForegroundColor','black',...
              'Title','',...
              'FontSize',14,...
              'TitlePosition','centertop',...
              'Position', [0.45 0.05 0.47 0.82]);

 % 'Movies available for the selected date:'
   uicontrol('Style', 'text', ...
             'Units', 'normalized', ...
             'FontSize', 14, ...
             'FontWeight', 'bold', ...
             'BackgroundColor', '#D6CADD', ...
             'Position', [0.485 0.79 0.4 0.05], ...
             'String', 'Movies available for the selected date:');

   %determine movies for the selected date
   if strcmp(selected_date, '01-01-2025')
       movies = {'10things.jpeg', 'crawdads.jpeg', 'watcher.jpeg', 'wicked.jpeg',
'scarymovie.jpeg', 'smile2.jpeg'};
   elseif strcmp(selected_date, '02-01-2025')
       movies = {'conductor.jpeg', 'homealone.jpeg', 'miraculos.jpeg', 'moana.jpeg',
'crawdads.jpeg', 'titanic.jpeg'};

         etc...
   else
       movies = {}; %no movies for other dates
   end
```

The part with the coment "%determine movies for the selected date" is having the working principle in the next way: with the comand if is compared if the variable selected_date is the same with '01-01-2025' if it is the same then the variable movies will be egual with the array of the movies {'10things.jpeg', 'crawdads.jpeg', 'watcher.jpeg', 'wicked.jpeg', 'scarymovie.jpeg', 'smile2.jpeg'}; and if not it is compared with the next date and so on until 9 February. If no date is matched then the variable movies will be egual with an empty array "movies = {};".

```
%if no movies are found for the selected date
    if isempty(movies)
        h = msgbox('No movies available for the selected date.', 'Warning', 'warn');
        desired_position = [550, 300, 220, 70];
        set(h, 'Position', desired_position);
        return;
    end
```

With the functin isempty it’s checked if the variable movies is empty or not. If movies is empty, the program executes the code inside the if comand where a message box with the warning 'No movies available for the selected date' is displayed . If it is not empty, it skips to this part of the program.

```
%get the selected genre from the dropdown
genre_dropdown = fig.UserData.genre_dropdown;
selected_option = genre_dropdown.String{genre_dropdown.Value};

 %filter movies based on the selected genre
if strcmp(selected_option, 'None')
    selected_movies = movies;
else
    selected_movies = filter_movies_by_genre(movies, selected_option);
end
```

Afterwards it is accesed the variable genre_dropdown from ig.UserData.genre_dropdown and with the genre_dropdown.String{genre_dropdown.Value} to get the genre of themovie. {genre_dropdown.Value} is giving me the index of the genre drop down menu.

With strcmp its compared the variable selected_option with „None” and if they are the same then selected_movies is assigned to movies which means that later on all the movies from the selected date will be displayed. If they are not the same then selected_movies is assigned the function filter_movies_by_genre(movies, selected_option) which will filter the movies in order to display only the movies of the selected genre.

```
%if no movies are found for the selected genre
    if isempty(selected_movies)
        h = msgbox('No movies available for the selected genre.', 'Warning', 'warn');
         set(h, 'Position', [550, 300, 220, 70]);
     else
         %display the selected movies
         for i = 1:numel(selected_movies)
             col = mod(i-1, 3);  %column (0, 1, 2)
             row = floor((i-1)/3);  %row (0, 1, 2, ...)
             %define position for each movie image
             ax = axes('Parent', group, ...
                  'Units', 'normalized', ...
                  'Position', [0.03 + 0.298*col, 0.5 - 0.43*row, 0.4, 0.4]);

             %load and display the image
             img = imread(selected_movies{i});
             imshow(img, 'Parent', ax);
         end
     end
 end
```

The code checks if the selected_movies variable is empty. If it is, a message box appears with a warning that no movies are available. Otherwise, the code calculates positions in a grid, creates axes, and displays the movie images within them using imshow.

```
function selected_movies = filter_movies_by_genre(movies, selected_option)
  if strcmp(selected_option, 'Animation')
    genre_movies = {'moana.jpeg', 'miraculos.jpeg'};
  elseif strcmp(selected_option, 'Action')
    genre_movies = {'fastandfurious.jpeg', 'spider_man.jpeg'};
  elseif strcmp(selected_option, 'Drama')
    genre_movies = {'wonder.jpeg', 'conductor.jpeg'};
  elseif strcmp(selected_option, 'Comedy')
    genre_movies = {'homealone.jpeg', 'scarymovie.jpeg'};
  elseif strcmp(selected_option, 'Horror')
    genre_movies = {'smile2.jpeg', 'orphan.jpeg'};
  elseif strcmp(selected_option, 'Romance')
    genre_movies = {'10things.jpeg', 'titanic.jpeg'};
  elseif strcmp(selected_option, 'Mystery')
    genre_movies = {'crawdads.jpeg', 'watcher.jpeg'};
  else
    genre_movies = {};  %if no genre matches
  end

  %filter movies for the selected genre and selected date
  selected_movies = intersect(movies, genre_movies);
end
```

In the function selected_movies = filter_movies_by_genre(movies, selected_option) the variable selected_option is compared to every movie genre from the drop down menu and if they match the genre_movies is assigned to the cell array with the specific movies. If it doesn’t match any movie genre then the genre_movies variable is assigned to an empty array.

Afterwards, the intersect function compares both arrays and returns a new array that contains only the elements present in **both arrays,** movies and genre_movies.

Because the function selected_movies = filter_movies_by_genre(movies, selected_option) returns a value (selected_movies) it has this form.). If the function did not need to return anything, the correct form would be function filter_movies_by_genre(movies, selected_option).

About the function buy_ticket(fig)

```
%add slider label
   uicontrol('Style', 'text', ...
          'Parent', fig, ...
          'Units', 'normalized', ...
          'Position', [0.01 0.85 0.2 0.05], ...
          'FontSize', 12, ...
          'FontWeight', 'bold', ...
          'BackgroundColor', '#FAF3E0', ...
          'String', 'Adjust Ticket Quantity:');
```

Here is created a tex button which will indicate to the user that it’s possible to adjust the quantity of tickets.

```
   %add slider
   slider = uicontrol('Style', 'slider', ...
                'Parent', fig, ...
                'Units', 'normalized', ...
                'Position', [0.01 0.8 0.2 0.05], ...
                'Min', 1, 'Max', 10, 'Value', 5, ...
                'SliderStep', [1/9, 1/9], ...
                'Callback', @(src, event) sliderCallback(src, event, fig));
```

Afterwards, it’s created the slider button

```
%add text to display slider value
   ticketDisplay = uicontrol('Style', 'text', ...
                     'Parent', fig, ...
                     'Units', 'normalized', ...
                     'Position', [0.01 0.75 0.2 0.05], ...
                     'FontSize', 12, ...
                     'BackgroundColor', '#FAF3E0', ...
                     'String', 'Tickets: 5');

fig.UserData.ticketDisplay = ticketDisplay;
   fig.UserData.selectedMovie = '';
   fig.UserData.enteredDate = '';
```

And there is also a button to display the slider’s value

fig.UserData.ticketDisplay = ticketDisplay is used to store the ticketDisplay inside fig.UserData so other functions can access and update the displayed number.

This initializes selectedMovie as an empty string in fig.UserData so that When the user selects a movie from the dropdown (moviesDropdown), the movieSelectionCallback function updates this value.

Like above, the enteredDate is initialized as an empty string in fig.UserData. When the user enters a date in the date input field, the dateInputCallback function updates this value.

```
moviesDropdown = uicontrol('Style', 'popupmenu', ...
                           'Units', 'normalized', ...
                           'Position', [0.01, 0.56, 0.22, 0.04], ...
                           'FontSize', 12, ...
                           'BackgroundColor', '#FFFFFF', ...
                           'String', {'Ten things I hate about you', 'Titanic', 'Where The Crawdads
Sing', 'Watcher', 'Lord of the rings', 'The wicked', 'The Conductor', 'Wonder ', 'Fast & Furious
9 ', 'Spider-Man ', 'Home Alone', 'Scary Movie ', 'Orphan ', 'Smile 2 '}, ...
                           'Enable', 'on', ...
                           'Callback', @(src, event) movieSelectionCallback(src, fig));

  uicontrol('Style', 'text', ...
        'Units', 'normalized', ...
        'FontSize', 12, ...
        'FontWeight', 'bold', ...
        'BackgroundColor', '#FAF3E0', ...
        'Position', [0.04 0.6 0.1 0.05], ...
        'String', 'Select movie:');

  uicontrol('Style', 'text', ...
        'Units', 'normalized', ...
        'FontSize', 12, ...
        'FontWeight', 'bold', ...
        'BackgroundColor', '#FAF3E0', ...
        'Position', [0.01 0.69 0.15 0.05], ...
        'String', 'Enter date:');

  dateInput = uicontrol('Style', 'edit', ...
                 'Parent', fig, ...
                 'Units', 'normalized', ...
                 'Position', [0.13 0.7 0.1 0.045], ...
                 'BackgroundColor', [0.9, 0.9, 0.9], ...
                 'ForegroundColor', 'black', ...
                 'FontSize', 12, ...
                 'String', '', ... % Start with an empty box
                 'Callback', @(src, ~) dateInputCallback(src, fig));

  uicontrol('Style','pushbutton',...
        'Parent', fig, ...
        'Units','normalized',...
        'FontSize', 14,...
        'BackgroundColor','#ADD8E6',...
        'Position',[0.01 0.2 0.2 0.1],...
        'String','Buy ticket', ...
        'Callback', @(src, event) buyTicketCallback(fig));
```

Then here are initialized :

- the dropdown meniu for the movies;

- a text button 'Select movie:';

- another text button 'Enter date:';

- a text button to enter date;

- variable dateInput which has the callback function @(src, ~) dateInputCallback(src, fig) meaning when the user types something in the text box, the dateInputCallback function is accesed;

- and a pushbutton.

```
most_watched_movies(fig);

end
```

And in the end of the function it’s called the function most_watched_movies(fig) in order to access that part

And from here there are multimple callbacks of functons.

```
function sliderCallback(src, ~, fig)
    currentValue = round(src.Value);
    fig.UserData.ticketDisplay.String = ['Tickets: ', num2str(currentValue)];
end
```

In sliderCallback the src.Value gets the current value of the slider which will be rounded to a integer value with round() in case the user enters a floating number and the ticketDisplay text box will be updated by fig.UserData.ticketDisplay.String

```
function movieSelectionCallback(src, fig)
    movies = src.String;
    selectedIndex = src.Value;
    fig.UserData.selectedMovie = movies{selectedIndex};
end
```

In the movieSelectionCallback, src.String and src.Value get the list of movie names available and the index of the currently selected movie from the dropdown so that movies{selectedIndex}will get the movie name corresponding to the selected index from the list of movies which will be stored in figure’s UserData with fig.UserData.selectedMovie so it can be used later.

```
function dateInputCallback(src, fig)
    % Update the entered date in UserData
    fig.UserData.enteredDate = src.String;
end
```

This callback updates the enteredDate property in the figure's UserData to store the date that the user enters in the text box. The date the user has entered in the text box is accesed with . src.String and it’s stored in UserData by fig.UserData.enteredDate.

```
function buyTicketCallback(fig)
    ticketDisplay = fig.UserData.ticketDisplay.String;
    numTickets = str2double(regexp(ticketDisplay, '\d+', 'match', 'once'));
    selectedMovie = fig.UserData.selectedMovie;
    enteredDate = fig.UserData.enteredDate;
    % Display confirmation message
    msg = sprintf('You bought %d tickets for %s for the movie %s', ...
                  numTickets, enteredDate, selectedMovie);
    msgbox(msg, 'Purchase Confirmation');
end
```

The current ticket count is accessesd from the ticketDisplay text box using fig.UserData.ticketDisplay.String. regexp(ticketDisplay, '\d+', 'match', 'once') extracts the numerical value from the text and str2double converts the extracted ticket count. fig.UserData.selectedMovie and fig.UserData.enteredDate gets the selected movie and entered date. A string is created (and it's alocated the variable msg) showing the user's purchase details, including the number of tickets, the selected movie, and the entered date. With msgbox a message box showing the confirmation message with the ticket purchase details.

About the function most_watched_movies(fig)

```
movies = {'Ten things I hate about you', 'Titanic', 'Where The Crawdads Sing', ...
          'Watcher', 'Lord of the rings', 'The wicked', 'The Conductor', ...
          'Wonder', 'Fast & Furious 9', 'Spider-Man', 'Home Alone', ...
          'Scary Movie', 'Orphan', 'Smile 2'};

days = {'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'};

fig.UserData.views = zeros(length(days), length(movies)); %rows for days, columns for
movies
fig.UserData.movies = movies;
fig.UserData.days = days;
```

In the beginning the movies variable is assigned to an array which contains every movie available and the days variable is assigned to anothr array which contains every days of the week.

fig.UserData.views = zeros(length(days), length(movies));

This line creates a matrix of zeros to represent the views for each movie on each day of the week. The length(days) returns the number of days from the days list, which is 7 (from Monday to Sunday).The length(movies) returns the number of movies in the movies list which is 14. So the first line creates a 7x14 matrix that is used to how many times each movie was watched on each of the seven days.

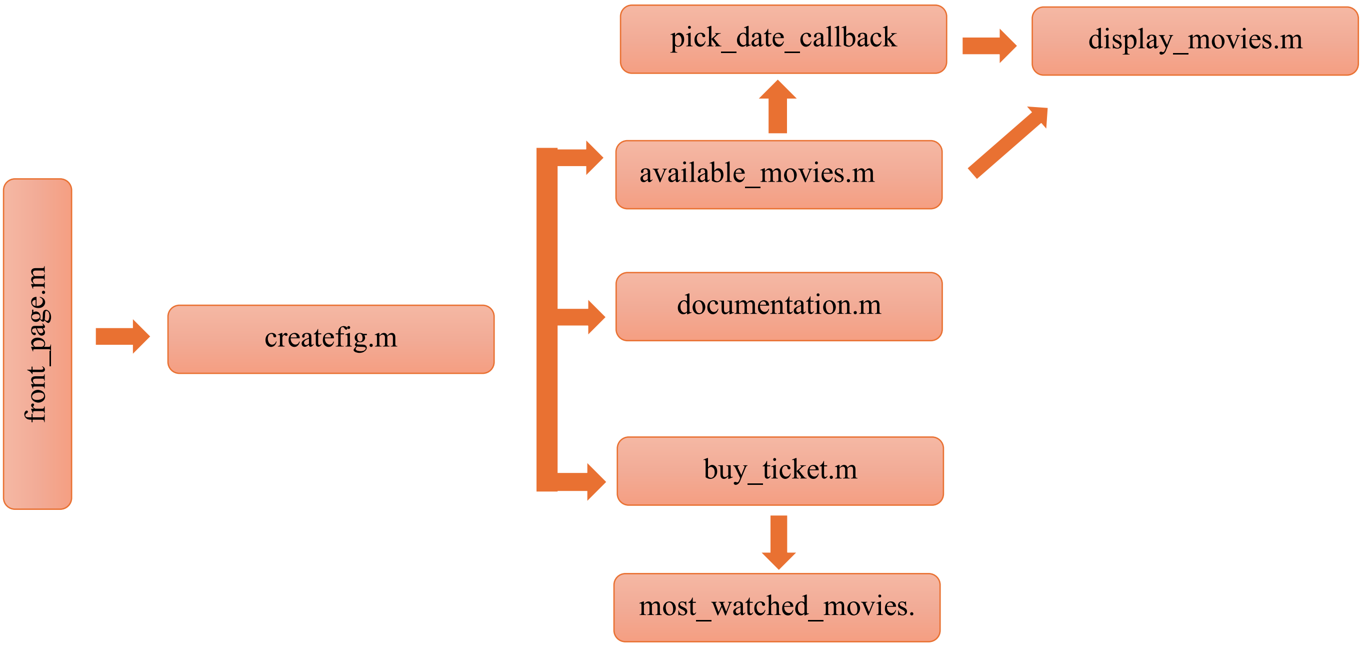
example

	Titanic	Smile 2	Spider-Man	etc
Monday	5	3	0	
Tuesday	0	0	7	
Wednseday	0	0	0	
etc				

Most-watched movies:
Monday: Titanic (5 views).
Tuesday: Spider-Man (7 views).
Wednesday: No views.

Afterwards 2 drop down menus to select the day and movie, a edit button for the number of views and 2 push buttons and 2 text buttons are created with uicontrol and implemented (some of them with anonymous functions).

BLOCK DIAGRAM



THE COMPLETE CODE

```
%main file front_page.m
close all;

figure ('Name','MOVIE TRACKING',...
       'Numbertitle', 'off',...
       'Color','#FAF3E0',...
       'Units','normalized',...
       'Position',[0.2 0.2 0.7 0.7]);
%title
uicontrol('Style','text', ...
         'Units','normalized', ...
         'Fontsize', 24, ...
         'Fontweight','bold', ...
         'BackgroundColor', 'white', ...
         'Position',[0.35 0.80 0.30 0.09], ...
         'String', 'MOVIE TRACKING');

%user menu
h = uimenu('Label', 'User_Menu');
uimenu(h, 'Label', 'Open Documentation', 'Callback', 'documentation');
uimenu(h, 'Label', 'Available Movies', 'Callback', 'avin=0;createfig(avin)');
uimenu(h, 'Label', 'Buy ticket', 'Callback', 'avin=1;createfig(avin)');
uimenu(h, 'Label', 'Display Picture1', 'Callback', ...
    'img = imread("pusheen.jpg"); figure("Name", "Pusheen", "NumberTitle", "off", "Color",
"#FFFFFF", "Units", "normalized", "Position", [0.2 0.2 0.6 0.6]); imshow(img, []);
title("Pusheen", "FontSize", 16, "FontWeight", "bold");');
uimenu(h, 'Label', 'Display Picture2', 'Callback', ...
    'img = imread("balaur.jpg"); figure("Name", "Balaur", "NumberTitle", "off", "Color",
"#FFFFFF", "Units", "normalized", "Position", [0.2 0.2 0.6 0.6]); imshow(img, []);
title("Balaur", "FontSize", 16, "FontWeight", "bold");');
uimenu(h, 'Label', 'Display Picture3', 'Callback', ...
    'img = imread("pugsheen.jpg"); figure("Name", "Pugsheen", "NumberTitle", "off", "Color",
"#FFFFFF", "Units", "normalized", "Position", [0.2 0.2 0.6 0.6]); imshow(img, []);
title("Pugsheen", "FontSize", 16, "FontWeight", "bold");');
uimenu(h, 'Label', 'Display Picture4', 'Callback', ...
    'img = imread("types.jpg"); figure("Name", "Pugsheen", "NumberTitle", "off", "Color",
"#FFFFFF", "Units", "normalized", "Position", [0.2 0.2 0.6 0.6]); imshow(img, []); title("",
"FontSize", 16, "FontWeight", "bold");');


%create axes for the background image
ax = axes('Position', [0 0 1 1]);
imshow('background.jpeg', 'Parent', ax);
```

```
%button to display the documentation
uicontrol('Style','pushbutton',...
        'Units','normalized',...
        'Fontsize', 14,...
        'BackgroundColor','#FFD1DF',...
        'Position',[0.4 0.6 0.20 0.1],...
        'String','Documentation',...
        'Callback','documentation');

%creates the figure for the Available movies option
uicontrol('Style','pushbutton',...
        'Units','normalized',...
        'Fontsize', 14,...
        'BackgroundColor','#FFD1DF',...
        'Position',[0.17 0.4 0.20 0.1],...
        'String','Available movies',...
        'Callback','avin=0;createfig(avin)');

%creates the figure for the Buy ticket option
uicontrol('Style','pushbutton',...
        'Units','normalized',...
        'Fontsize', 14,...
        'BackgroundColor','#FFD1DF',...
        'Position',[0.63 0.4 0.20 0.1],...
        'String','Buy ticket',...
        'Callback','avin=1;createfig(avin)');

%close button
uicontrol('Style','pushbutton',...
        'Units','normalized',...
        'Fontsize', 12,...
        'BackgroundColor','#D6CADD',...
        'Position',[0.82 0.03 0.1 0.06],...
        'string','Close',...
        'Callback','close');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function createfig(avin, fig)

 %if the user pushed the "Available movies" button, the title of the figure becomes "Available
movies" and variable avin=0
   if avin == 0
       figurename = "Available movies";
   else
       % if the user pushed the "Buy ticket" button, the title of the figure becomes "Buy ticket"
and variable avin=1
```

```
        figurename = "Buy ticket";
    end

    %create the figure with the specified title
    fig = figure('Name', figurename, ...
                'NumberTitle', 'off', ...
                'Color', '#FAF3E0', ...
                'Units', 'normalized', ...
                'Position', [0.1 0.1 0.7 0.7]);

    %add a title to the figure
    uicontrol('Style', 'text', ...
              'Units', 'normalized', ...
              'FontSize', 16, ...
              'FontWeight', 'bold', ...
              'BackgroundColor', '#FAF3E0', ...
              'Position', [0.3 0.9 0.4 0.1], ...
              'String', figurename);

    if avin == 0
    available_movies(fig);
    else
    buy_ticket(fig);
    end
end

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

function available_movies(fig)

uibuttongroup('Visible','on',...
              'BackgroundColor','#D6CADD',...
              'ForegroundColor','black',...
              'Title','',...
              'FontSize',14,...
              'TitlePosition','centertop',...
              'Position', [ 0.05 0.44 0.28 0.5]);

    %'Select Movie Date:'
    uicontrol('Style', 'text', ...
              'Units', 'normalized', ...
              'FontSize', 14, ...
              'FontWeight', 'bold', ...
              'BackgroundColor', '#D6CADD', ...
              'Position', [0.1 0.74 0.2 0.15], ...
              'String', 'Select Movie Date:');
```

```
%button to pick a date
uicontrol('Style', 'pushbutton', ...
        'Units', 'normalized', ...
        'Fontsize', 14, ...
        'BackgroundColor', '#ADD8E6', ...
        'Position', [0.1 0.8 0.2 0.05], ...
        'String', 'Pick date', ...
        'Callback', @(src, event) pick_date_callback(src, event, fig));

%'Date selected:'
uicontrol('Style', 'text', ...
        'Units', 'normalized', ...
        'FontSize', 14, ...
        'FontWeight', 'bold', ...
        'BackgroundColor', '#D6CADD', ...
        'Position', [0.1 0.735 0.2 0.05], ...
        'String', 'Date selected:');

%'No date selected'
date_display = uicontrol('Style', 'text', ...
                    'Units', 'normalized', ...
                    'FontSize', 14, ...
                    'FontWeight', 'normal', ...
                    'BackgroundColor', '#D6CADD', ...
                    'Position', [0.1 0.7 0.2 0.05], ...
                    'String', 'No date selected   ');

%store date_display handle in the figure's UserData
fig.UserData.date_display = date_display;

%'Select Movie Genre:'
 uicontrol('Style', 'text', ...
        'Units', 'normalized', ...
        'FontSize', 14, ...
        'FontWeight', 'bold', ...
        'BackgroundColor', '#D6CADD', ...
        'Position', [0.1 0.50 0.2 0.15], ...
        'String', 'Select Movie Genre:');

 %dropdown menu
   genre_dropdown = uicontrol('Style', 'popupmenu', ...
                         'Units', 'normalized', ...
                         'Position', [0.1, 0.55, 0.2, 0.04], ...
                         'FontSize', 12, ...
                         'BackgroundColor', '#FFFFFF', ...
                         'String', {'None', 'Animation', 'Action', 'Drama', 'Comedy', 'Horror',
'Romance', 'Mystery'}, ...
                         'Enable', 'off', ...
                         'Callback', @(src, event) genre_selection_callback(src, event, fig));

      %store genre_dropdown handle in the figure's UserData
      fig.UserData.genre_dropdown = genre_dropdown;
   end

%genre selection callback
function genre_selection_callback(src, event, fig)
   %get the current date from the figure
   date_display = fig.UserData.date_display;
   selected_date = date_display.String; %get the date string

   %if a date is selected, get the selected genre
   genres = src.String;
   selected_option = genres{src.Value};%get the selected genre

   %call the display_movies function
   display_movies(fig, selected_date);
end

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

function pick_date_callback(~, ~, fig)
   %create a new figure for the date picker
   date_picker_fig = uifigure('Name', 'Pick a Date', ...
                       'NumberTitle', 'off', ...
                       'Color', '#FAF3E0', ...
                       'Units', 'normalized', ...
                       'Position', [0.1 0.13 0.3 0.3]);

   %add a date picker widget to the new figure
   uidatepicker(date_picker_fig, 'DisplayFormat', 'dd-MM-yyyy', ...
             'Position', [100 210 150 22], ...
             'Value', datetime(2025, 1, 1), ...
             'ValueChangedFcn', @(dp, event) date_change(dp, event, fig));
end

%callback function for the date picker
function date_change(dp, ~, fig)
   selected_date = datestr(dp.Value, 'dd-mm-yyyy'); %get the selected date

   %update the text control with the selected date
   date_display = fig.UserData.date_display;
```

```
    date_display.String = [' ', selected_date];

    %genre dropdown-on after a date is selected
    genre_dropdown = fig.UserData.genre_dropdown;
    genre_dropdown.Enable = 'on';

    display_movies(fig, selected_date);
end

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

function display_movies(fig, selected_date)

   group=uibuttongroup('Visible','on',...
              'BackgroundColor','#D6CADD',...
              'ForegroundColor','black',...
              'Title','',...
              'FontSize',14,...
              'TitlePosition','centertop',...
              'Position', [0.45 0.05 0.47 0.82]);

 %'Movies available for the selected date:'
    uicontrol('Style', 'text', ...
            'Units', 'normalized', ...
            'FontSize', 14, ...
            'FontWeight', 'bold', ...
            'BackgroundColor', '#D6CADD', ...
            'Position', [0.485 0.79 0.4 0.05], ...
            'String', 'Movies available for the selected date:');

    %determine movies for the selected date
    if strcmp(selected_date, '01-01-2025')
        movies = {'10things.jpeg', 'crawdads.jpeg', 'watcher.jpeg', 'wicked.jpeg',
'scarymovie.jpeg', 'smile2.jpeg'};
    elseif strcmp(selected_date, '02-01-2025')
        movies = {'conductor.jpeg', 'homealone.jpeg', 'miraculos.jpeg', 'moana.jpeg',
'crawdads.jpeg', 'titanic.jpeg'};
    elseif strcmp(selected_date, '03-01-2025')
        movies = {'fastandfurious.jpeg', 'orphan.jpeg', 'lordoftherings.jpeg', 'smile2.jpeg',
'wicked.jpeg', 'titanic.jpeg'};
    elseif strcmp(selected_date, '04-01-2025')
        movies = {'10things.jpeg', 'wonder.jpeg', 'homealone.jpeg', 'spider_man.jpeg',
'miraculos.jpeg', 'smile2.jpeg'};
    elseif strcmp(selected_date, '05-01-2025')
        movies = {'smile2.jpeg', 'titanic.jpeg', 'miraculos.jpeg', 'homealone.jpeg', 'wicked.jpeg',
'watcher.jpeg'};
```

```
    elseif strcmp(selected_date, '06-01-2025')
        movies = {'10things.jpeg', 'spider_man.jpeg', 'orphan.jpeg', 'smile2.jpeg', 'titanic.jpeg',
'wonder.jpeg'};
    elseif strcmp(selected_date, '07-01-2025')
        movies = {'titanic.jpeg', 'wicked.jpeg', 'spider_man.jpeg', 'conductor.jpeg', 'smile2.jpeg',
'watcher.jpeg'};
    elseif strcmp(selected_date, '08-01-2025')
        movies = {'10things.jpeg', 'miraculos.jpeg', 'fastandfurious.jpeg', 'conductor.jpeg',
'homealone.jpeg', 'smile2.jpeg'};
    elseif strcmp(selected_date, '09-01-2025')
        movies = {'spider_man.jpeg', 'homealone.jpeg', 'wicked.jpeg', 'miraculos.jpeg',
'smile2.jpeg', 'titanic.jpeg'};
    elseif strcmp(selected_date, '10-01-2025')
        movies = {'wicked.jpeg', 'miraculos.jpeg', 'homealone.jpeg', '10things.jpeg',
'conductor.jpeg', 'watcher.jpeg'};
    elseif strcmp(selected_date, '11-01-2025')
        movies = {'miraculos.jpeg', 'wicked.jpeg', 'smile2.jpeg', '10things.jpeg',
'fastandfurious.jpeg', 'conductor.jpeg'};
    elseif strcmp(selected_date, '12-01-2025')
        movies = {'smile2.jpeg', 'spider_man.jpeg', 'wicked.jpeg', '10things.jpeg',
'conductor.jpeg', 'watcher.jpeg'};
    elseif strcmp(selected_date, '13-01-2025')
        movies = {'orphan.jpeg', 'homealone.jpeg', 'wicked.jpeg', '10things.jpeg',
'miraculos.jpeg', 'fastandfurious.jpeg'};
    elseif strcmp(selected_date, '14-01-2025')
        movies = {'homealone.jpeg', 'miraculos.jpeg', 'smile2.jpeg', '10things.jpeg',
'watcher.jpeg', 'conductor.jpeg'};
    elseif strcmp(selected_date, '15-01-2025')
        movies = {'watcher.jpeg', 'smile2.jpeg', 'fastandfurious.jpeg', 'orphan.jpeg',
'10things.jpeg', 'wicked.jpeg'};
    elseif strcmp(selected_date, '16-01-2025')
        movies = {'conductor.jpeg', 'smile2.jpeg', '10things.jpeg', 'wicked.jpeg', 'miraculos.jpeg',
'fastandfurious.jpeg'};
    elseif strcmp(selected_date, '17-01-2025')
        movies = {'fastandfurious.jpeg', 'wicked.jpeg', 'smile2.jpeg', '10things.jpeg',
'homealone.jpeg', 'watcher.jpeg'};
    elseif strcmp(selected_date, '18-01-2025')
        movies = {'10things.jpeg', 'wonder.jpeg', 'homealone.jpeg', 'fastandfurious.jpeg',
'miraculos.jpeg', 'smile2.jpeg'};
    elseif strcmp(selected_date, '19-01-2025')
        movies = {'smile2.jpeg', 'titanic.jpeg', 'miraculos.jpeg', 'homealone.jpeg', 'wicked.jpeg',
'watcher.jpeg'};
    elseif strcmp(selected_date, '20-01-2025')
        movies = {'10things.jpeg', 'fastandfurious.jpeg', 'orphan.jpeg', 'smile2.jpeg', 'titanic.jpeg',
'wonder.jpeg'};
    elseif strcmp(selected_date, '21-01-2025')
```

```
        movies = {'titanic.jpeg', 'wicked.jpeg', 'fastandfurious.jpeg', 'conductor.jpeg',
'smile2.jpeg', 'watcher.jpeg'};
    elseif strcmp(selected_date, '22-01-2025')
        movies = {'10things.jpeg', 'miraculos.jpeg', 'fastandfurious.jpeg', 'conductor.jpeg',
'homealone.jpeg', 'smile2.jpeg'};
    elseif strcmp(selected_date, '23-01-2025')
        movies = {'spider_man.jpeg', 'homealone.jpeg', 'wicked.jpeg', 'miraculos.jpeg',
'smile2.jpeg', 'titanic.jpeg'};
    elseif strcmp(selected_date, '24-01-2025')
        movies = {'wicked.jpeg', 'miraculos.jpeg', 'homealone.jpeg', '10things.jpeg',
'conductor.jpeg', 'watcher.jpeg'};
    elseif strcmp(selected_date, '25-01-2025')
        movies = {'miraculos.jpeg', 'wicked.jpeg', 'smile2.jpeg', '10things.jpeg',
'fastandfurious.jpeg', 'conductor.jpeg'};
    elseif strcmp(selected_date, '26-01-2025')
        movies = {'smile2.jpeg', 'fastandfurious.jpeg', 'wicked.jpeg', '10things.jpeg',
'conductor.jpeg', 'watcher.jpeg'};
    elseif strcmp(selected_date, '27-01-2025')
        movies = {'orphan.jpeg', 'homealone.jpeg', 'wicked.jpeg', '10things.jpeg',
'miraculos.jpeg', 'fastandfurious.jpeg'};
    elseif strcmp(selected_date, '28-01-2025')
        movies = {'homealone.jpeg', 'miraculos.jpeg', 'smile2.jpeg', '10things.jpeg',
'watcher.jpeg', 'conductor.jpeg'};
    elseif strcmp(selected_date, '29-01-2025')
        movies = {'watcher.jpeg', 'smile2.jpeg', 'spider_man.jpeg', 'orphan.jpeg', '10things.jpeg',
'wicked.jpeg'};
    elseif strcmp(selected_date, '30-01-2025')
        movies = {'conductor.jpeg', 'smile2.jpeg', '10things.jpeg', 'wicked.jpeg', 'miraculos.jpeg',
'fastandfurious.jpeg'};
    elseif strcmp(selected_date, '31-01-2025')
        movies = {'fastandfurious.jpeg', 'wicked.jpeg', 'smile2.jpeg', '10things.jpeg',
'homealone.jpeg', 'watcher.jpeg'};
    elseif strcmp(selected_date, '01-02-2025')
        movies = {'10things.jpeg', 'wonder.jpeg', 'homealone.jpeg', 'fastandfurious.jpeg',
'miraculos.jpeg', 'smile2.jpeg'};
    elseif strcmp(selected_date, '02-02-2025')
        movies = {'smile2.jpeg', 'titanic.jpeg', 'miraculos.jpeg', 'homealone.jpeg', 'wicked.jpeg',
'watcher.jpeg'};
    elseif strcmp(selected_date, '03-02-2025')
        movies = {'10things.jpeg', 'fastandfurious.jpeg', 'orphan.jpeg', 'smile2.jpeg', 'titanic.jpeg',
'wonder.jpeg'};
    elseif strcmp(selected_date, '04-02-2025')
        movies = {'titanic.jpeg', 'wicked.jpeg', 'fastandfurious.jpeg', 'conductor.jpeg',
'smile2.jpeg', 'watcher.jpeg'};
    elseif strcmp(selected_date, '05-02-2025')
```

```
        movies = {'10things.jpeg', 'miraculos.jpeg', 'spider_man.jpeg', 'conductor.jpeg',
'homealone.jpeg', 'smile2.jpeg'};
    elseif strcmp(selected_date, '06-02-2025')
        movies = {'fastandfurious.jpeg', 'homealone.jpeg', 'wicked.jpeg', 'miraculos.jpeg',
'smile2.jpeg', 'titanic.jpeg'};
    elseif strcmp(selected_date, '07-02-2025')
        movies = {'wicked.jpeg', 'miraculos.jpeg', 'homealone.jpeg', '10things.jpeg',
'conductor.jpeg', 'watcher.jpeg'};
    elseif strcmp(selected_date, '08-02-2025')
        movies = {'miraculos.jpeg', 'wicked.jpeg', 'smile2.jpeg', '10things.jpeg',
'fastandfurious.jpeg', 'conductor.jpeg'};
    elseif strcmp(selected_date, '09-02-2025')
        movies = {'smile2.jpeg', 'spider_man.jpeg', 'wicked.jpeg', '10things.jpeg',
'conductor.jpeg', 'watcher.jpeg'};
    else
        movies = {}; %no movies for other dates
    end

    %if no movies are found for the selected date
    if isempty(movies)
        h = msgbox('No movies available for the selected date.', 'Warning', 'warn');
        desired_position = [550, 300, 220, 70];
        set(h, 'Position', desired_position);
        return;
    end

    %get the selected genre from the dropdown
    genre_dropdown = fig.UserData.genre_dropdown;
    selected_option = genre_dropdown.String{genre_dropdown.Value};

     %filter movies based on the selected genre
    if strcmp(selected_option, 'None')
        selected_movies = movies;
    else
        selected_movies = filter_movies_by_genre(movies, selected_option);
    end

       %if no movies are found for the selected genre
       if isempty(selected_movies)
           h = msgbox('No movies available for the selected genre.', 'Warning', 'warn');
            set(h, 'Position', [550, 300, 220, 70]);
        else
            %display the selected movies
            for i = 1:numel(selected_movies)
                col = mod(i-1, 3);  % Column (0, 1, 2)
                row = floor((i-1)/3);  % Row (0, 1, 2, ...)
            %define position for each movie image
            ax = axes('Parent', group, ...
                   'Units', 'normalized', ...
                   'Position', [0.03 + 0.298*col, 0.5 - 0.43*row, 0.4, 0.4]);

            %load and display the image
            img = imread(selected_movies{i});
            imshow(img, 'Parent', ax);
         end
      end
end

function selected_movies = filter_movies_by_genre(movies, selected_option)

   if strcmp(selected_option, 'Animation')
      genre_movies = {'moana.jpeg', 'miraculos.jpeg'};
   elseif strcmp(selected_option, 'Action')
      genre_movies = {'fastandfurious.jpeg', 'spider_man.jpeg'};
   elseif strcmp(selected_option, 'Drama')
      genre_movies = {'wonder.jpeg', 'conductor.jpeg'};
   elseif strcmp(selected_option, 'Comedy')
      genre_movies = {'homealone.jpeg', 'scarymovie.jpeg'};
   elseif strcmp(selected_option, 'Horror')
      genre_movies = {'smile2.jpeg', 'orphan.jpeg'};
   elseif strcmp(selected_option, 'Romance')
      genre_movies = {'10things.jpeg', 'titanic.jpeg'};
   elseif strcmp(selected_option, 'Mystery')
      genre_movies = {'crawdads.jpeg', 'watcher.jpeg'};
   else
      genre_movies = {};  %if no genre matches
   end

   %filter movies
   selected_movies = intersect(movies, genre_movies);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function buy_ticket(fig)

 % Add slider label
   uicontrol('Style', 'text', ...
          'Parent', fig, ...
          'Units', 'normalized', ...
          'Position', [0.01 0.85 0.2 0.05], ...
          'FontSize', 12, ...
```

```
            'FontWeight', 'bold', ...
            'BackgroundColor', '#FAF3E0', ...
            'String', 'Adjust Ticket Quantity:');

    % Add slider
    slider = uicontrol('Style', 'slider', ...
                  'Parent', fig, ...
                  'Units', 'normalized', ...
                  'Position', [0.01 0.8 0.2 0.05], ...
                  'Min', 1, 'Max', 10, 'Value', 5, ...
                  'SliderStep', [1/9, 1/9], ...
                  'Callback', @(src, event) sliderCallback(src, event, fig));

    % Add text to display slider value
    ticketDisplay = uicontrol('Style', 'text', ...
                      'Parent', fig, ...
                      'Units', 'normalized', ...
                      'Position', [0.01 0.75 0.2 0.05], ...
                      'FontSize', 12, ...
                      'BackgroundColor', '#FAF3E0', ...
                      'String', 'Tickets: 5');

     % Store display handle in figure UserData for updating
    fig.UserData.ticketDisplay = ticketDisplay;
    fig.UserData.selectedMovie = '';
    fig.UserData.enteredDate = '';

    moviesDropdown = uicontrol('Style', 'popupmenu', ...
                        'Units', 'normalized', ...
                        'Position', [0.01, 0.56, 0.22, 0.04], ...
                        'FontSize', 12, ...
                        'BackgroundColor', '#FFFFFF', ...
                        'String', {'Ten things I hate about you', 'Titanic', 'Where The Crawdads
Sing', 'Watcher', 'Lord of the rings', 'The wicked', 'The Conductor', 'Wonder ', 'Fast & Furious
9 ', 'Spider-Man ', 'Home Alone', 'Scary Movie ', 'Orphan ', 'Smile 2 '}, ...
                        'Enable', 'on', ...
                        'Callback', @(src, event) movieSelectionCallback(src, fig));

    uicontrol('Style', 'text', ...
            'Units', 'normalized', ...
            'FontSize', 12, ...
            'FontWeight', 'bold', ...
            'BackgroundColor', '#FAF3E0', ...
            'Position', [0.04 0.6 0.1 0.05], ...
            'String', 'Select movie:');
    uicontrol('Style', 'text', ...
            'Units', 'normalized', ...
            'FontSize', 12, ...
            'FontWeight', 'bold', ...
            'BackgroundColor', '#FAF3E0', ...
            'Position', [0.01 0.69 0.15 0.05], ...
            'String', 'Enter date:');

     % Text box for date input
    dateInput = uicontrol('Style', 'edit', ...
                    'Parent', fig, ...
                    'Units', 'normalized', ...
                    'Position', [0.13 0.7 0.1 0.045], ...
                    'BackgroundColor', [0.9, 0.9, 0.9], ...
                    'ForegroundColor', 'black', ...
                    'FontSize', 12, ...
                    'String', '', ... % Start with an empty box
                    'Callback', @(src, ~) dateInputCallback(src, fig));

    uicontrol('Style','pushbutton',...
            'Parent', fig, ...
            'Units','normalized',...
            'FontSize', 14,...
            'BackgroundColor','#ADD8E6',...
            'Position',[0.01 0.2 0.2 0.1],...
            'String','Buy ticket', ...
            'Callback', @(src, event) buyTicketCallback(fig));

most_watched_movies(fig);

end

function sliderCallback(src, ~, fig)
    % Update the ticket display with the rounded slider value
    currentValue = round(src.Value);
    fig.UserData.ticketDisplay.String = ['Tickets: ', num2str(currentValue)];
end

function movieSelectionCallback(src, fig)
    % Update the selected movie in UserData
    movies = src.String;
    selectedIndex = src.Value;
    fig.UserData.selectedMovie = movies{selectedIndex};
end

function dateInputCallback(src, fig)
```

```
    % Update the entered date in UserData
    fig.UserData.enteredDate = src.String;
end

function buyTicketCallback(fig)
    % Get the necessary data
    ticketDisplay = fig.UserData.ticketDisplay.String;
    numTickets = str2double(regexp(ticketDisplay, '\d+', 'match', 'once'));
    selectedMovie = fig.UserData.selectedMovie;
    enteredDate = fig.UserData.enteredDate;

    % Display confirmation message
    msg = sprintf('You bought %d tickets for %s for the movie %s', ...
                  numTickets, enteredDate, selectedMovie);
    msgbox(msg, 'Purchase Confirmation');
end

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

function most_watched_movies(fig)
    movies = {'Ten things I hate about you', 'Titanic', 'Where The Crawdads Sing', ...
              'Watcher', 'Lord of the rings', 'The wicked', 'The Conductor', ...
              'Wonder', 'Fast & Furious 9', 'Spider-Man', 'Home Alone', ...
              'Scary Movie', 'Orphan', 'Smile 2'};

    days = {'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'};

    %matrix initializtion
    fig.UserData.views = zeros(length(days), length(movies)); %rows for days, columns for
movies
    fig.UserData.movies = movies;
    fig.UserData.days = days;

    uibuttongroup('Visible', 'on', ...
                  'BackgroundColor', '#D6CADD', ...
                  'ForegroundColor', 'black', ...
                  'Title', '', ...
                  'FontSize', 14, ...
                  'TitlePosition', 'centertop', ...
                  'Position', [0.45 0.05 0.47 0.82]);

    %'Select day and movie:'
    uicontrol('Style', 'text', ...
              'Units', 'normalized', ...
              'FontSize', 14, ...
              'FontWeight', 'bold', ...
```

```
        'BackgroundColor', '#D6CADD', ...
        'Position', [0.485 0.79 0.4 0.05], ...
        'String', 'Select day and movie:');

%day dropdown
days_dropdown = uicontrol('Style', 'popupmenu', ...
                   'Units', 'normalized', ...
                   'Position', [0.47, 0.75, 0.13, 0.04], ...
                   'FontSize', 12, ...
                   'BackgroundColor', '#FFFFFF', ...
                   'String', days, ...
                   'Enable', 'on', ...
                   'Callback', @(src, ~) day_selection_callback(src, fig));

fig.UserData.days_dropdown = days_dropdown;

%movie dropdown
movies_dropdown = uicontrol('Style', 'popupmenu', ...
                    'Units', 'normalized', ...
                    'Position', [0.64, 0.75, 0.2, 0.04], ...
                    'FontSize', 12, ...
                    'BackgroundColor', '#FFFFFF', ...
                    'String', movies, ...
                    'Enable', 'on', ...
                    'Callback', @(src, ~) movie_selection_callback(src, fig));

%'Number of views'
uicontrol('Style', 'text', ...
        'Units', 'normalized', ...
        'Position', [0.47 0.67 0.12 0.05], ...
        'BackgroundColor', '#D6CADD', ...
        'ForegroundColor', 'black', ...
        'FontSize', 12, ...
        'String', 'Number of Views:');

%edit object for entering number of views
uicontrol('Style', 'edit', ...
        'Units', 'normalized', ...
        'Position', [0.6 0.685 0.05 0.045], ...
        'BackgroundColor', [0.9, 0.9, 0.9], ...
        'ForegroundColor', 'black', ...
        'FontSize', 12, ...
        'String', ", ...
        'Callback', @(src, ~) views_callback(src, fig));

%button 'Show Most Watched Movies'
```

```
    uicontrol('Style', 'pushbutton', ...
            'Units', 'normalized', ...
            'Position', [0.47, 0.6, 0.4, 0.05], ...
            'BackgroundColor', '#ADD8E6', ...
            'ForegroundColor', 'black', ...
            'FontSize', 14, ...
            'String', 'Show Most Watched Movies', ...
            'Callback', @(~, ~) display_results(fig));

    %button 'Plot Most Watched Movies'
    uicontrol('Style', 'pushbutton', ...
            'Units', 'normalized', ...
            'Position', [0.47, 0.52, 0.4, 0.05], ...
            'BackgroundColor', '#ADD8E6', ...
            'ForegroundColor', 'black', ...
            'FontSize', 14, ...
            'String', 'Plot Most Watched Movies', ...
            'Callback', @(~, ~) plot_results(fig));
end

%callback for Day Selection
function day_selection_callback(src, fig)
    selected_day = src.String{src.Value}; %get selected_day
    fig.UserData.selected_day = selected_day; % store selected day
    disp(['Day selected: ', selected_day]);
end

%callback for Movie Selection
function movie_selection_callback(src, fig)
    selected_movie = src.String{src.Value}; %get selected_movie
    fig.UserData.selected_movie = selected_movie; %store selected movie
    disp(['Movie selected: ', selected_movie]);
end

%callback for Number of Views
function views_callback(src, fig)
    input_value = str2double(src.String); %convert input to number

    %retrieve selected day and movie
    day_index = find(strcmp(fig.UserData.days, fig.UserData.selected_day));%find index of
the day in the array
    movie_index = find(strcmp(fig.UserData.movies, fig.UserData.selected_movie));%same
for movie

    %update views for the selected day and movie
    fig.UserData.views(day_index, movie_index) = input_value;
    disp(['Views updated: ', num2str(input_value), ' for ', fig.UserData.selected_movie, ' on ',
fig.UserData.selected_day]);
end

%display results
function display_results(fig)
    views = fig.UserData.views;
    movies = fig.UserData.movies;
    days = fig.UserData.days;

    %display most-watched movie for each day
    results = ";%empty string
    for i = 1:size(views, 1) %i=day_index
        [max_views, max_index] = max(views(i, :));
        if max_views > 0
            results = [results, days{i}, ': ', movies{max_index}, ...
                        ' with ', num2str(max_views), ' views.'];
        else
            results = [results, days{i}, ': No views recorded.'];
        end
    end

    %show results in a dialog box
    msgbox(results, 'Most Watched Movies', 'help');
end

%plot results
function plot_results(fig)
    views = fig.UserData.views;
    movies = fig.UserData.movies;
    days = fig.UserData.days;

    %find the most-watched movie and its views for each day
    most_watched_views = zeros(size(views, 1), 1);
    most_watched_movies = cell(size(views, 1), 1);

    for i = 1:size(views, 1)
        [most_watched_views(i), max_index] = max(views(i, :));
        if most_watched_views(i) > 0
            most_watched_movies{i} = movies{max_index};
        else
            most_watched_movies{i} = 'No views';
        end
    end
    %create bar chart
    figure('Position', [100, 100, 1000, 600]);
    bar(most_watched_views,'FaceColor', '#FFD1DF');
    set(gca, 'XTickLabel', days, 'XTick', 1:length(days));
    xlabel('Days');
    ylabel('Number of Views');
    title('Most Watched Movies per Day');
    grid on;

    %add movie names as text labels above bars
    for i = 1:length(most_watched_views)
        text(i, most_watched_views(i) + 0.5, most_watched_movies{i}, ...
             'HorizontalAlignment', 'center', 'FontSize', 10, 'Color', 'black');
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% function file documentation.m

function documentation
open('Documentation.pdf');
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

The end

BIBLOGRAPHY

Book “MATLAB for Students”, vol 1 by Mihaela Cîrlugea and Paul Farago, UTPRESS
Book “MATLAB for Students”, vol 2 by Mihaela Cîrlugea and Paul Farago, UTPRESS
https://en.wikipedia.org/wiki/MATLAB
https://www.geeksforgeeks.org/how-to-create-a-dropdown-menu-in-matlab/
https://www.tutorialspoint.com/how-to-create-a-dropdown-menu-in-matlab
https://ch.mathworks.com/matlabcentral/answers/2104901-how-to-make-a-dropdown-menu
https://ch.mathworks.com/help/matlab/matlab_prog/anonymous-functions.html
https://ch.mathworks.com/help/matlab/ref/uidatepicker.html
https://ch.mathworks.com/help/matlab/ref/uidatepicker.html
https://ch.mathworks.com/help/matlab/ref/uidatepicker.html
https://www.youtube.com/watch?v=3bFKyiSLG9o
https://www.geeksforgeeks.org/anonymous-functions-in-matlab/
https://ch.mathworks.com/matlabcentral/answers/2059394-function-that-makes-anonymous-functions
+ more