

FAZAIA BILQUIS COLLEGE OF EDUCATION
FOR WOMEN PAF BASE NUR KHAN
COMPUTER SCIENCE DEPARTMENT



NUMERICAL ANALYSIS AND COMPUTATION
LAB PROJECT REPORT

PROJECT TITLE:

ALUMNI MANAGEMENT SYSTEM

SUBMITTED TO

Ms. Saadia Noor

SUBMITTED BY

- **Aafreen Zahra Kazmi (37)**
- **Fatima Tariq (55)**
- **Alia Mahreez (52)**
- **Arooj Khan (81)**
- **Laiba Fatima (62)**
- **Rubina Shaheen (45)**

SEMESTER / SECTION

- **5th B**

SUBMISSION DATE

- **10 - Dec - 2024**

Contents

1. Project Name	1
2. Introduction.....	1
3. Objective	1
4. Key Points	1
5. Daily Life Example	1
6. Tools Used.....	1
7. Advantages.....	2
8. Future Works	2
9. Coding Section.....	2
<input type="checkbox"/> Initialization and Main Menu	2
<input type="checkbox"/> Adding Alumni Data.....	3
<input type="checkbox"/> Search by Profession.....	5
<input type="checkbox"/> Search by Graduation.....	6
<input type="checkbox"/> Delete Alumni Data.....	7
<input type="checkbox"/> Display Summary	7
<input type="checkbox"/> Export Data	8
10. Output Section.....	9
11. Conclusion	15

1. Project Name

Alumni Management System

2. Introduction

This project involves developing a GUI-based application to manage alumni data effectively. The system allows users to add, search, delete, and view alumni details, making it a centralized repository for alumni records.

3. Objective

- To design a user-friendly system for managing alumni data.
 - To provide features for adding, searching, and deleting alumni information.
 - To generate summaries and export data for reporting purposes.
-

4. Key Points

- **Add Alumni Data:** Input and save alumni details, including profile pictures.
 - **Search Alumni:** Search records based on profession or graduation year.
 - **Delete Alumni:** Remove outdated or incorrect records.
 - **Export Data:** Save records to an external file for sharing or backup.
 - **Data Visualization:** View summaries like profession distribution using graphs.
-

5. Daily Life Example

Universities and colleges often need a database of their alumni for:

- Organizing alumni meetups.
 - Tracking professional achievements of graduates.
 - Sending updates and newsletters.
-

6. Tools Used

- **Programming Language:** MATLAB
- **Database:** Microsoft Excel
- **GUI Development:** MATLAB's App Designer and figure-based GUI tools.

7. Advantages

- Centralized alumni data storage.
 - Simplified search and retrieval.
 - Easy maintenance and updates.
 - Data export for external use.
 - User-friendly graphical interface.
-

8. Future Works

- Integration with an online database for cloud-based storage.
 - Adding more search filters and advanced analytics.
 - Automated email notifications to alumni.
 - Mobile application support.
-

9. Coding Section

• Initialization and Main Menu

```
function alumniManagementSystem()
    % Main GUI setup
    close all;
    global excelFile;
    excelFile = fullfile('C:\Users\AAFREEN ZAHRA KAZMI\Desktop', 'alumniData1.xlsx'); % Path to Excel file
    initializeDatabase(); % Ensure the database exists
    createMainInterface(); % Create the main menu
end

function initializeDatabase()
    % Initialize the database if it doesn't exist
    global excelFile;
    if exist(excelFile, 'file') ~= 2 % Check if the file exists
        headers = {'Name', 'GraduationYear', 'Profession', 'Email', 'PhoneNumber', 'ProfilePicture'}; % Removed ID
        xlswrite(excelFile, headers, 'Sheet1', 'A1'); % Write headers to Excel file
    end
end

function createMainInterface()
    % Create the main interface
    f = figure('Name', 'Alumni Management System', 'NumberTitle', 'off', ...
        'Position', [300, 150, 800, 600], 'MenuBar', 'none', ...
        'Color', [0.8 0.9 1], 'Resize', 'on');

    % Title
    uicontrol('Style', 'text', 'String', 'Alumni Management System', ...
        'FontSize', 16, 'FontWeight', 'bold', ...
```

```

        'Position', [200, 500, 400, 40], 'BackgroundColor', [0.8 0.9 1]);

% Buttons for modules
uicontrol('Style', 'pushbutton', 'String', '1. Add Alumni Data', ...
    'Position', [250, 400, 200, 50], 'FontSize', 12, ...
    'Callback', @(~, ~) addDataInterface(f));

uicontrol('Style', 'pushbutton', 'String', '2. Search Alumni by Profession', ...
    'Position', [250, 350, 200, 50], 'FontSize', 12, ...
    'Callback', @(~, ~) searchByProfessionInterface(f));

uicontrol('Style', 'pushbutton', 'String', '3. Search Alumni by Graduation Year', ...
    'Position', [250, 300, 200, 50], 'FontSize', 12, ...
    'Callback', @(~, ~) searchGraduationYearInterface(f));

uicontrol('Style', 'pushbutton', 'String', '4. Delete Alumni Data', ...
    'Position', [250, 250, 200, 50], 'FontSize', 12, ...
    'Callback', @(~, ~) deleteDataInterface(f));

uicontrol('Style', 'pushbutton', 'String', '5. Display Alumni Summary', ...
    'Position', [250, 200, 200, 50], 'FontSize', 12, ...
    'Callback', @(~, ~) displaySummary());

uicontrol('Style', 'pushbutton', 'String', '6. Export Data to File', ...
    'Position', [250, 150, 200, 50], 'FontSize', 12, ...
    'Callback', @(~, ~) exportData());

% Close button
uicontrol('Style', 'pushbutton', 'String', 'Close', ...
    'Position', [250, 50, 200, 50], 'FontSize', 12, ...
    'Callback', @(~, ~) close(f));
End

```

• Adding Alumni Data

```

function addDataInterface(parentFig)
% Interface for adding alumni data
if isvalid(parentFig), close(parentFig); end
f = figure('Name', 'Add Alumni Data', 'NumberTitle', 'off', ...
    'Position', [300, 150, 800, 600], 'Color', [0.9 0.9 1], 'Resize', 'on');

uicontrol('Style', 'text', 'String', 'Enter Alumni Details:', ...
    'FontSize', 14, 'Position', [250, 520, 200, 30], ...
    'BackgroundColor', [0.9 0.9 1]);

% Input fields
fields = {'Name', 'Graduation Year', 'Profession', 'Email', 'Phone Number'};
yPos = 450:-50:200;
inputs = cell(size(fields));
for i = 1:length(fields)
    uicontrol('Style', 'text', 'String', [fields{i}, ':'], ...
        'Position', [100, yPos(i), 120, 25], 'FontSize', 10, 'BackgroundColor', [0.9 0.9 1]);
end

```

```

        inputs{i} = uicontrol('Style', 'edit', 'Position', [230, yPos(i), 300, 30], 'FontSize', 10);
    end

    % Picture upload section
    uicontrol('Style', 'text', 'String', 'Upload Picture:', ...
        'Position', [550, 350, 120, 25], 'FontSize', 10, 'BackgroundColor', [0.9 0.9 1]);
    uicontrol('Style', 'pushbutton', 'String', 'Upload', ...
        'Position', [550, 300, 100, 30], 'FontSize', 12, ...
        'Callback', @(~, ~) uploadPicture(f));

    % Placeholder for the picture
    ax = axes('Position', [0.75, 0.2, 0.2, 0.4]);

    % Submit button
    uicontrol('Style', 'pushbutton', 'String', 'Submit', ...
        'Position', [350, 50, 100, 40], 'FontSize', 12, ...
        'Callback', @(~, ~) saveUserData(inputs, ax, f));

    % Close button
    uicontrol('Style', 'pushbutton', 'String', 'Close', ...
        'Position', [50, 50, 100, 40], 'FontSize', 12, ...
        'Callback', @(~, ~) close(f));

    % Previous button
    uicontrol('Style', 'pushbutton', 'String', 'Previous', ...
        'Position', [150, 50, 100, 40], 'FontSize', 12, ...
        'Callback', @(~, ~) createMainInterface());
end

function uploadPicture(f)
    [fileName, filePath] = uigetfile({'*.jpg;*.jpeg;*.png', 'Image Files (*.jpg, *.jpeg, *.png)'});
    if fileName
        ax = findobj(f, 'Type', 'axes');
        img = imread(fullfile(filePath, fileName));
        imshow(img, 'Parent', ax);
    end
end

function saveUserData(inputs, ax, f)
    % Save user data to Excel
    global excelFile;

    % Read existing data
    [~, ~, existingData] = xlsread(excelFile);

    % Prepare new data
    newData = cell(1, size(existingData, 2)); % Ensure consistent dimensions
    for i = 1:length(inputs)
        newData{i} = inputs{i}.String;
    end

    % Handle profile picture
    img = getimage(ax);
    if ~isempty(img)

```

```

imgPath = fullfile('C:\Users\AAFREEN ZAHRA KAZMI\Desktop', 'profile_picture.jpg');
imwrite(img, imgPath);
newData{end} = imgPath;
else
    newData{end} = "; % Empty if no picture uploaded
end

% Write data to Excel
existingData = [existingData; newData];
xlswrite(excelFile, existingData, 'Sheet1');

% Notify the user and reset interface
msgbox('Alumni data saved successfully');
close(f);
createMainInterface();
end

```

• Search by Profession

```

function searchByProfessionInterface(parentFig)
% Interface for searching alumni by profession
if isvalid(parentFig), close(parentFig); end
f = figure('Name', 'Search by Profession', 'NumberTitle', 'off', ...
    'Position', [300, 150, 800, 600], 'Color', [0.9 0.9 1], 'Resize', 'on');

uicontrol('Style', 'text', 'String', 'Enter Profession to Search:', ...
    'FontSize', 12, 'Position', [200, 450, 200, 30], 'BackgroundColor', [0.9 0.9 1]);
professionInput = uicontrol('Style', 'edit', 'Position', [400, 450, 200, 30]);

uicontrol('Style', 'pushbutton', 'String', 'Search', ...
    'Position', [350, 400, 100, 30], ...
    'Callback', @(~, ~) searchByProfession(professionInput, f)); % Profession is in column 3

% Close button
uicontrol('Style', 'pushbutton', 'String', 'Close', ...
    'Position', [50, 50, 100, 40], 'FontSize', 12, ...
    'Callback', @(~, ~) close(f));

% Previous button
uicontrol('Style', 'pushbutton', 'String', 'Previous', ...
    'Position', [150, 50, 100, 40], 'FontSize', 12, ...
    'Callback', @(~, ~) createMainInterface());
end

function searchByProfession(professionInput, f)
% Search alumni by profession
global excelFile;
profession = professionInput.String;
[~, ~, data] = xlsread(excelFile);

% Find matching entries
matchData = {};

```



```

for i = 2:size(data, 1)
    if contains(data{i, 3}, profession, 'IgnoreCase', true)
        matchData = [matchData; data(i, :)];
    end
end

% Display the results
if isempty(matchData)
    msgbox('No matching alumni found.');
```

else

```

    figure;
    t = uitable('Data', matchData, 'ColumnName', {'Name', 'GraduationYear', 'Profession', 'Email', 'PhoneNumber',
'ProfilePicture'}, ...
        'Position', [100, 100, 600, 400]);
end
end
end
```

- Search by Graduation

```

function searchGraduationYearInterface(parentFig)
    % Interface for searching by graduation year
    if isValid(parentFig), close(parentFig); end
    f = figure('Name', 'Search by Graduation Year', 'NumberTitle', 'off', ...
        'Position', [300, 150, 800, 600], 'Color', [0.9 0.9 1], 'Resize', 'on');

    uicontrol('Style', 'text', 'String', 'Enter Graduation Year to Search:', ...
        'FontSize', 12, 'Position', [200, 450, 200, 30], 'BackgroundColor', [0.9 0.9 1]);
    graduationYearInput = uicontrol('Style', 'edit', 'Position', [400, 450, 200, 30]);

    uicontrol('Style', 'pushbutton', 'String', 'Search', ...
        'Position', [350, 400, 100, 30], ...
        'Callback', @(~, ~) searchByGraduationYear(graduationYearInput, f));

    % Close button
    uicontrol('Style', 'pushbutton', 'String', 'Close', ...
        'Position', [50, 50, 100, 40], 'FontSize', 12, ...
        'Callback', @(~, ~) close(f));

    % Previous button
    uicontrol('Style', 'pushbutton', 'String', 'Previous', ...
        'Position', [150, 50, 100, 40], 'FontSize', 12, ...
        'Callback', @(~, ~) createMainInterface());
end

function searchByGraduationYear(graduationYearInput, f)
    % Search alumni by graduation year
    global excelFile;
    year = str2double(graduationYearInput.String);
    [~, ~, data] = xlsread(excelFile);

    % Find matching entries
    matchData = {};
    for i = 2:size(data, 1)
        if str2double(data{i, 2}) == year
```

```

        matchData = [matchData; data(i, :)];
    end
end

% Display the results
if isempty(matchData)
    msgbox('No matching alumni found.');
```

else

```

    figure;
    t = uitable('Data', matchData, 'ColumnName', {'Name', 'GraduationYear', 'Profession', 'Email', 'PhoneNumber',
'ProfilePicture'}, ...
        'Position', [100, 100, 600, 400]);
    end
end
```

• Delete Alumni Data

```

function deleteData(nameInput, f)
    % Delete alumni data
    global excelFile;
    name = nameInput.String;
    [~, ~, data] = xlsread(excelFile);

    % Find matching entry and delete it
    matchIndex = [];
    for i = 2:size(data, 1)
        if strcmpi(data{i, 1}, name)
            matchIndex = [matchIndex, i];
        end
    end

    if isempty(matchIndex)
        msgbox('No matching alumni found to delete.');
```

else

```

        data(matchIndex, :) = []; % Remove matching rows
        xlswrite(excelFile, data, 'Sheet1'); % Write back the updated data
        msgbox('Alumni data deleted successfully.');
```

end

```

end
```

• Display Summary

```

function displaySummary()
    % Display the summary of alumni data
    global excelFile;
    [~, ~, data] = xlsread(excelFile);

    % Calculate total number of alumni
    totalAlumni = size(data, 1) - 1; % Exclude header row
    professions = unique(data(2:end, 3)); % Unique professions
```

```

% Display total number of alumni
msgbox(['Total number of alumni: ', num2str(totalAlumni)]);

% Create a bar graph for profession distribution
professionCounts = zeros(size(professions));
for i = 1:length(professions)
    professionCounts(i) = sum(strcmp(data(:, 3), professions{i}));
end

figure;
bar(professionCounts);
set(gca, 'XTickLabel', professions);
title('Alumni Profession Distribution');
xlabel('Profession');
ylabel('Count');
end

```

- Export Data

```

function exportData()
% Export the alumni data to a text file
global excelFile;

% Read the data from Excel
[~, ~, data] = xlsread(excelFile);

if isempty(data)
    msgbox('No alumni data available');
else
    % Open a file dialog to choose the save location
    [fileName, filePath] = uiputfile('*.txt', 'Save Data As');

    if fileName
        try
            % Open the text file for writing
            fileID = fopen(fullfile(filePath, fileName), 'w');

            % Write headers (column names)
            headers = {'Name', 'GraduationYear', 'Profession', 'Email', 'PhoneNumber', 'ProfilePicture'};
            fprintf(fileID, '%s\t', headers{:});
            fprintf(fileID, '\n');

            % Write the data
            for i = 1:size(data, 1)
                fprintf(fileID, '%s\t', data{i, :});
                fprintf(fileID, '\n');
            end

            % Close the file
            fclose(fileID);

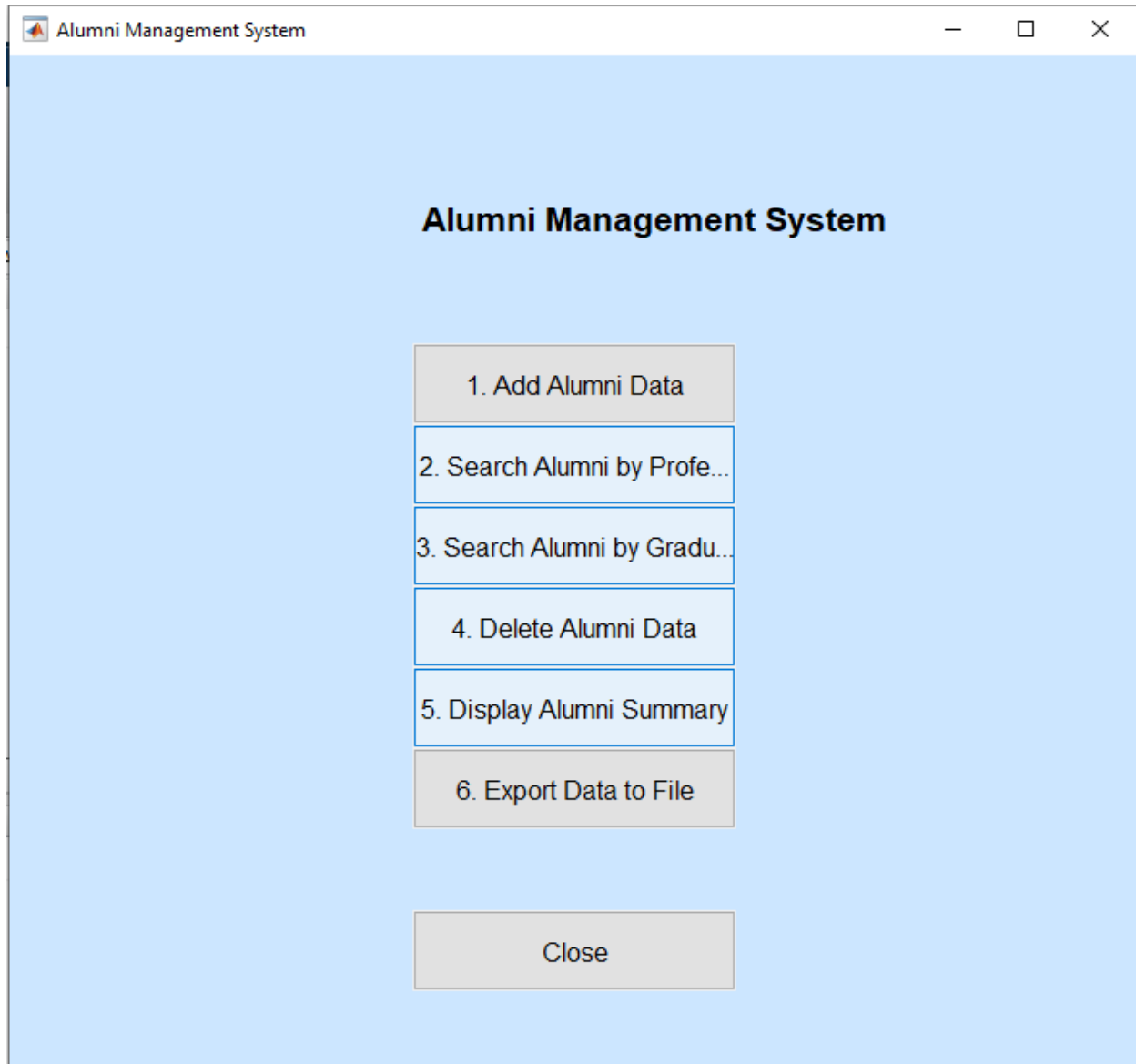
            % Show success message
            msgbox('Data exported successfully');
        catch ME
    end
end

```

```
        msgbox(['Error exporting data: ', ME.message]);  
    end  
end  
end  
end
```

10. Output Section

- **Menu section**



- **Adding Alumni Data:** Displays a success message once data is added.

Add Alumni Data

File Edit View Insert Tools Desktop Window Help

Enter Alumni Details:


Name:

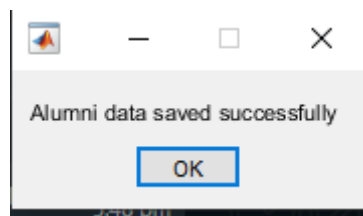
Graduation Year:

Profession:

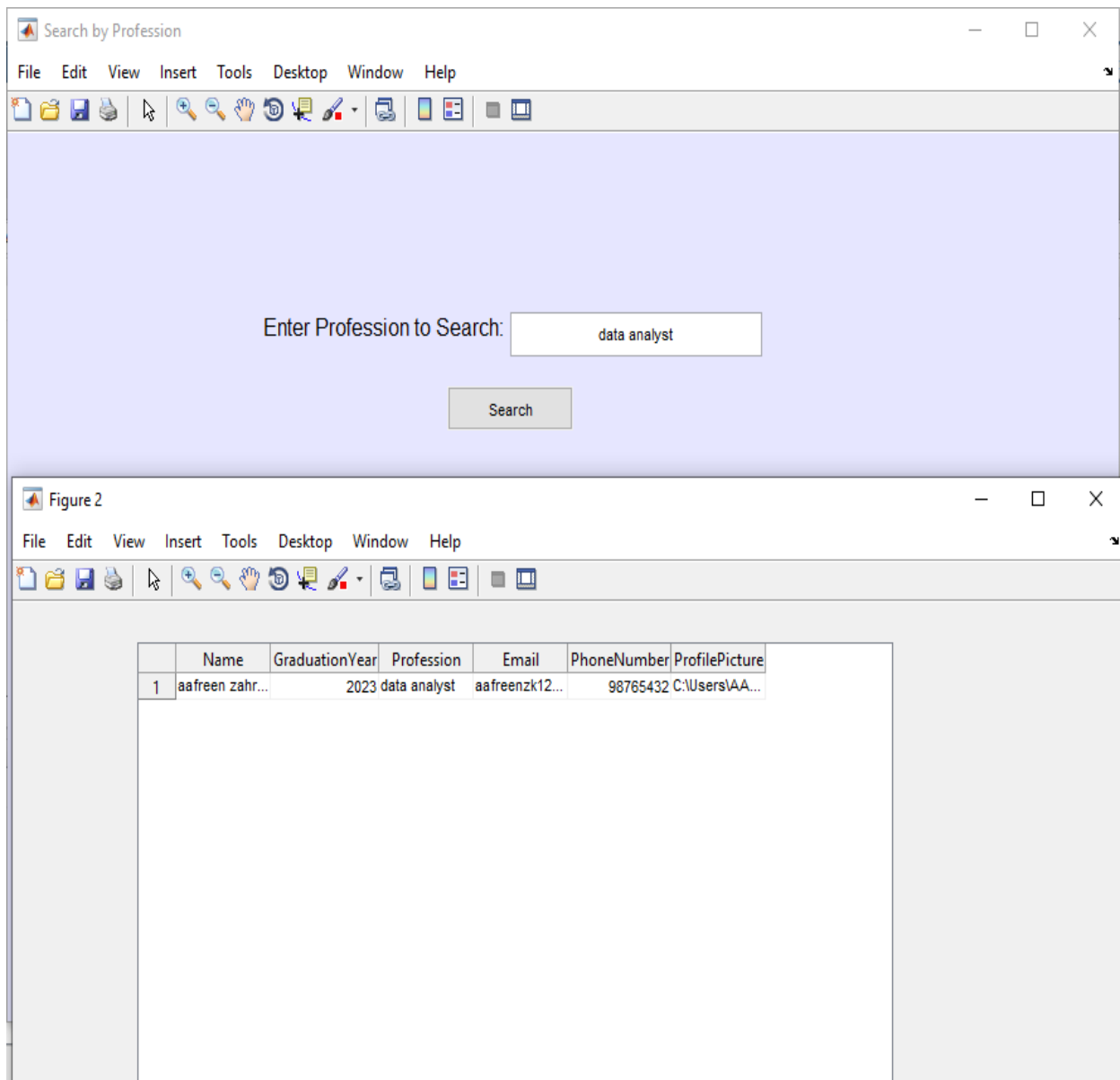
Email:

Phone Number:

Upload Picture: 



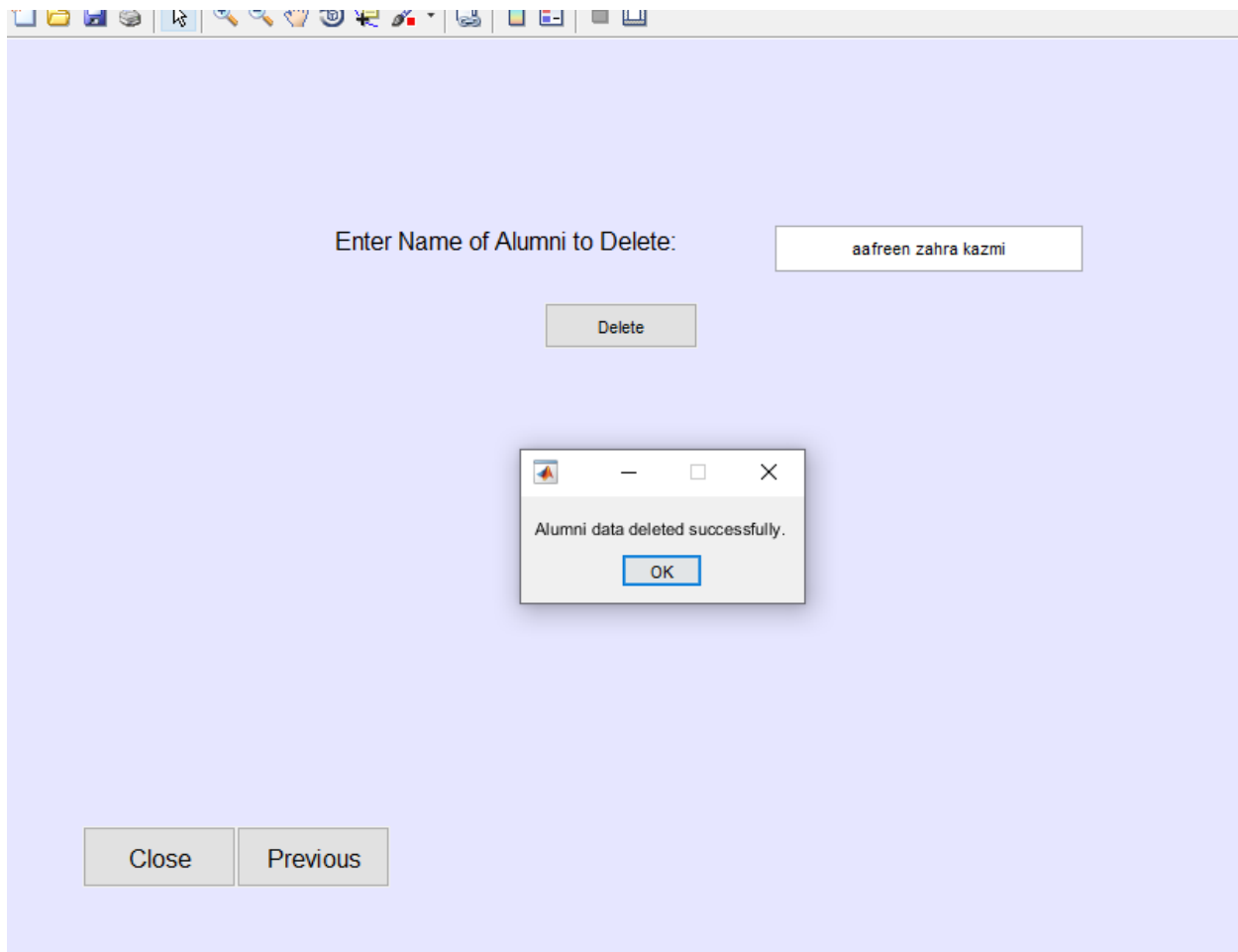
- **Search Results:** Shows a table with matching alumni records.
 - **Search by profession**



- **Search by Graduation Year**

The screenshot shows a web application window with a light blue background. At the top, there is a toolbar with various icons. Below the toolbar, the text "Enter Graduation Year to Search" is displayed. To the right of this text is a text input field containing the value "2021". Below the input field is a "Search" button. In the center of the window, a small modal dialog box is open, displaying the message "No matching alumni found." and an "OK" button. At the bottom left of the window, there are two buttons labeled "Close" and "Previous".

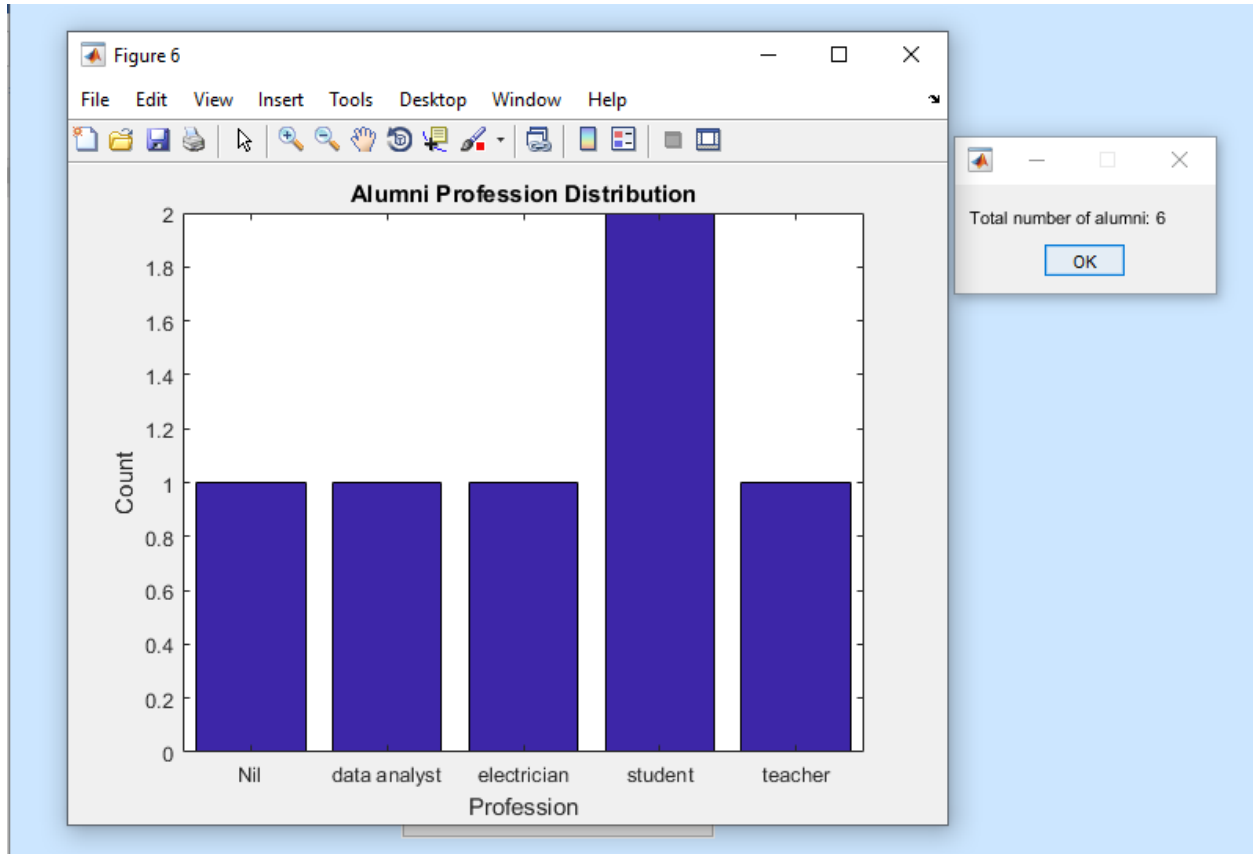
- **Delete Alumni Data:** Notifies if data is successfully deleted or not found.



The screenshot shows a web application interface with a light blue background. At the top, there is a toolbar with various icons. The main content area contains a form with the following elements:

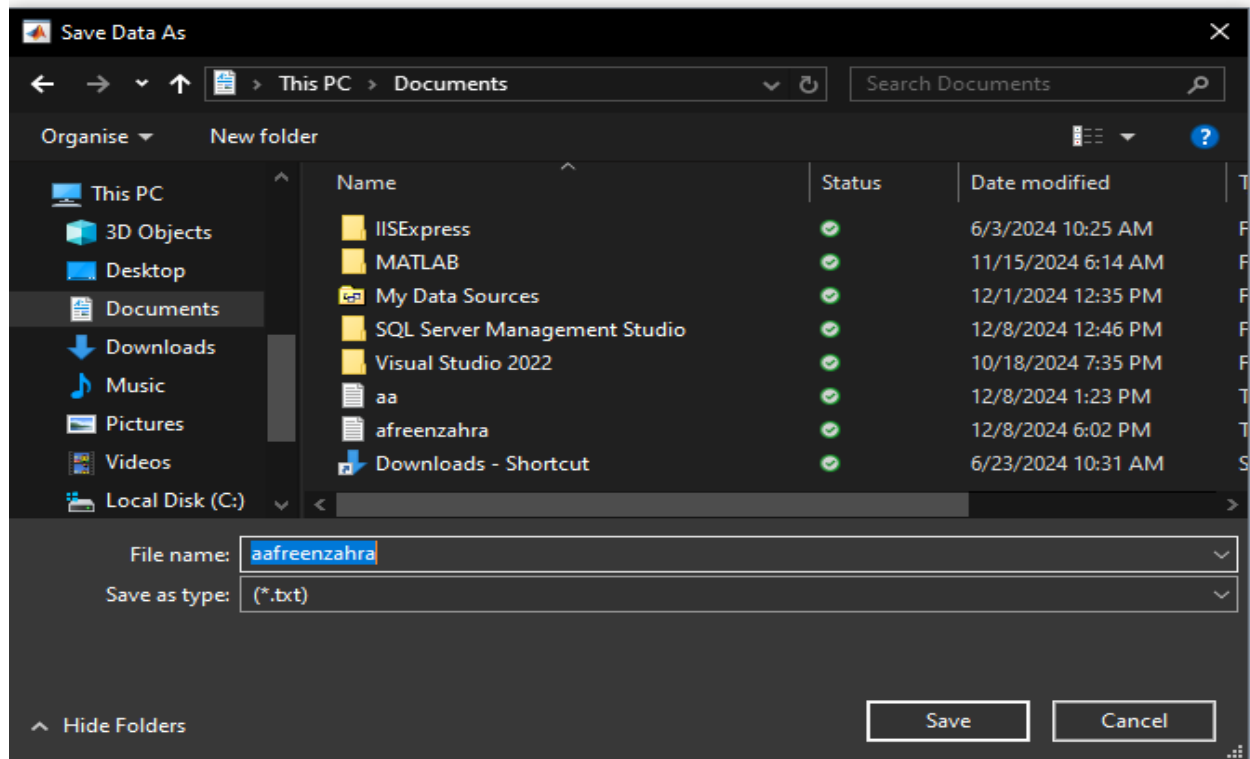
- A label "Enter Name of Alumni to Delete:" followed by a text input field containing the name "aafreen zahra kazmi".
- A "Delete" button below the input field.
- A modal dialog box in the center of the screen with the title "Alumni data deleted successfully." and an "OK" button.
- At the bottom left, there are two buttons: "Close" and "Previous".

- **Summary:** Displays the total count of alumni and a bar graph for profession distribution.



- **Export Data:** Exports data to a specified CSV file with a confirmation message.

Alumni Management System



alumniData1 - Excel (Product Activation Failed)

	A	B	C	D	E	F	G	H	I	J
1	Name	GraduationYear	Profession	Email	PhoneNumber	ProfilePicture				
2	fatima	2022	student	jadiuw	657	C:\Users\AAFREEN ZAHRA KAZMI\Desktop\profile_picture.jpg				
3	abdullah	2012	student	jhduwd	73654632	C:\Users\AAFREEN ZAHRA KAZMI\Desktop\profile_picture.jpg				
4	fatima tariq	2024	Nil	fatima@1234gmail.com	657237982	C:\Users\AAFREEN ZAHRA KAZMI\Desktop\profile_picture.jpg				
5	shaukat abbas kazmi	2024	electrician	shaukarkazmi@gmail.com	7364409128	C:\Users\AAFREEN ZAHRA KAZMI\Desktop\profile_picture.jpg				
6	shahaab kazmi	2014	teacher	jahdu@gmail.com	8465	C:\Users\AAFREEN ZAHRA KAZMI\Desktop\profile_picture.jpg				
7	aafreen zahra kazmi	2023	data analyst	aafreenzk1214@gmail.com	98765432	C:\Users\AAFREEN ZAHRA KAZMI\Desktop\profile_picture.jpg				
8										
9										
10										

11. Conclusion

The Alumni Management System simplifies alumni data management with an intuitive GUI, robust data handling, and user-friendly functionalities. It enhances operational efficiency and offers a scalable solution for alumni data management needs.