



PYTHON AND SQL SERVER
BASED

.....

BASEBALL ORGANIZATION **MANAGEMENT SYSTEM**

Set BY
PETROS ASEGID
MAKBEL KEBEDE
KIDIST YITBAREK

Let's Get Started





INTRODUCTION

A desktop-based system to manage a baseball organization

Built using Python, Tkinter, and SQL Server

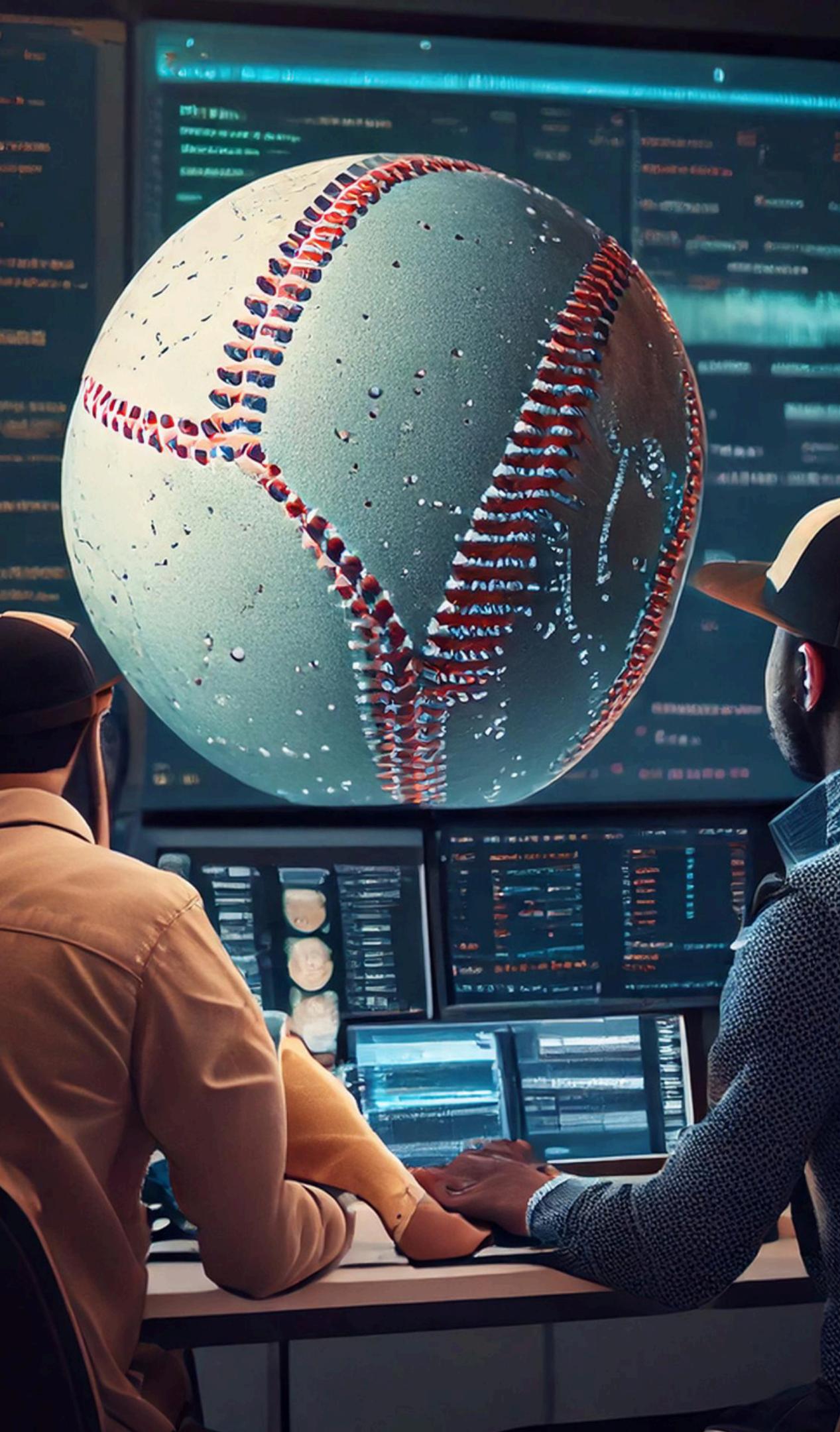
- Supports Personnel management, team management, and game management



Baseball organization Case Study

Consider a database system for a baseball organization such as the major leagues. The data requirements are summarized as follows:

- The personnel involved in the league include players, coaches, managers, and umpires. Each is identified by a unique personnel id. They are also described by their first and last names along with the date and place of birth.
- Players are further described by other attributes such as their batting orientation (left, right, or switch) and have a lifetime batting average (BA).
- Within the players group is a subset of players called pitchers. Pitchers have a lifetime ERA (earned run average) associated with them.
- Teams are uniquely identified by their names. Teams are also described by the city in which they are located and the division and league in which they play (such as Central division of the American League).





Teams have one manager, a number of coaches, and a number of players.

- Games are played between two teams, with one designated as the home team and the other the visiting team on a particular date. The score (runs, hits, and errors) is recorded for each team. The team with the most runs is declared the winner of the game.
- With each finished game, a winning pitcher and a losing pitcher are recorded. In case there is a save awarded, the save pitcher is also recorded.
- With each finished game, the number of hits (singles, doubles, triples, and home runs) obtained by each player is also recorded. Design an enhanced entity-relationship diagram for the BASEBALL data base and enter the design using a data modeling tool such as ERwin or Rational Rose.



MAIN FEATURES



Personnel Management

In a Baseball Organization Management System, personnel management would include managing coaches, players, Umpires, and other team Managers, tracking their Information,



Team Management

Personnel Management involves creating teams, assigning players and staff, and displaying team information for efficient organization and performance tracking.



Game Management

Game Management involves scheduling games, adding game results, and displaying result history for performance analysis and tracking.





TECHNOLOGY STACK

● PYTHON

● DATABASE

● GUI

● FILE

TKINTER- PYTHON
LIBRARY

FOR SAVING ADMIN
PASSWORD



Welcome

Admin

Fans



WELCOME PAGE

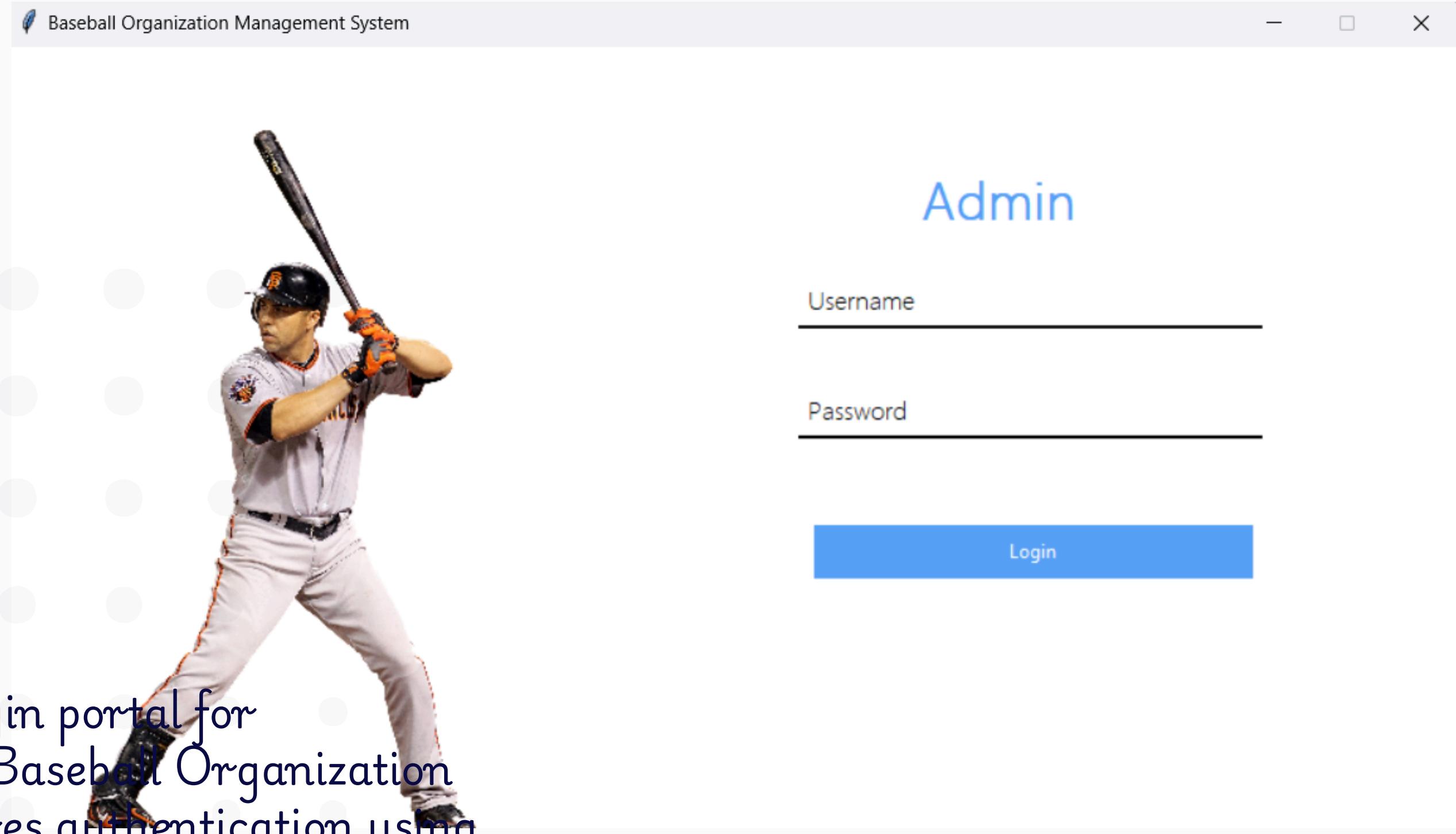
The Welcome Page serves as the entry point to the Baseball Organization Management System, providing users with options to access different functionalities. It features an intuitive interface with buttons for Admin Login and Fan Access, ensuring seamless navigation and user engagement.





ADMIN PAGE

The Admin Page is a secure login portal for administrators to manage the Baseball Organization Management System. It requires authentication using a username and password to ensure authorized access. Upon successful login, administrators can oversee team management, personnel assignments, game scheduling, and result tracking, ensuring smooth system operations.



Baseball Organization Management System

Admin

Username

Password

Login



```
def check_login():

    try:
        with open("credentials.txt", "r") as file:
            saved_username = file.readline().strip()
            saved_password = file.readline().strip()
    except FileNotFoundError:
        messagebox.showerror("Error", "Credentials file not found.")
        return

    # Get the entered username and password
    entered_username = user.get()
    entered_password = Password.get()

    # Check if the credentials match
    if entered_username == saved_username and entered_password == saved_password:
        Editorial_page() # Open the Hello page on successful login
    else:
        error_label.config(text="Incorrect username or password", fg="red")
```



SOME SNIPPETS

```
6 import pypyodbc as odbc
7
8 DRIVER_NAME = 'SQL Server'
9 SERVER_NAME = 'PETERASE'
10 DATABASE_NAME = 'Baseball organization'
11
12 # Correct Connection String
13 connection_string = f"""
14     DRIVER={{{{DRIVER_NAME}}}};
15     SERVER={{{SERVER_NAME}}};
16     DATABASE={{{DATABASE_NAME}}};
17     Trust_Connection=yes;
18 """
19
20 try:
21     conn = odbc.connect(connection_string)
22     print("Connection Successful:", conn)
23 except odbc.Error as e:
24     print("Connection Failed:", e)
```

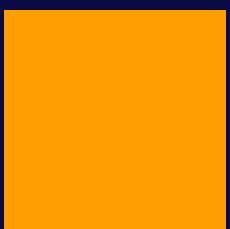


The Editorial Page allows administrators to manage and update important information within the Baseball Organization

Management System. It provides tools to edit team details, update game records, modify personnel assignments, and oversee key system data, ensuring accurate and up-to-date management of the organization.

The screenshot shows a Windows application window titled "Baseball Organization Management System". The window has a menu bar with "Personnel Management", "Team Management", and "Game Management". Below the menu is a toolbar with buttons for "Add Personnel", "View Personnel", "Update Personnel Data", and "Delete Personnel". The main area contains four text input fields labeled "First Name", "Last Name", "Date of Birth", and "Place of Birth", each with a corresponding empty input box. Below these is a dropdown menu labeled "Role" with the option "Select Role". At the bottom right of the input area is a blue button labeled "Add Personnel".

EDITORIAL PAGE



.....

CHALLENGES FACED

- Implementing secure login with file handling
- - Ensuring smooth database integration with SQL Server
- - Designing an easy-to-use GUI with Tkinter