**Project**

**Student Name:** Saransh Arora **UID:** 24MCI10041

**Branch:** MCA(AIML) **Section/Group:** 1(B)

**Semester:** 1 **Date of Performance:** 20-10-2024

**Subject Name:** PL/SQL LAB **Subject Code:** 24CAP-602

## **Introduction**

In today’s digital world, user authentication and management are crucial components of many applications. This project aims to develop a simple user management system that allows users to register and log in using a graphical user interface (GUI) built with Python's Tkinter library. The system uses MySQL as the database to store user credentials securely.

## **Acknowledgements**

This project leverages the capabilities of Python's Tkinter for GUI development and MySQL for backend data storage. Special thanks to the maintainers of these libraries and to the broader programming community for their contributions and support.

## **Problem Statement**

The need for effective user management systems is growing as applications increasingly require user authentication. Many existing solutions are complex, not user-friendly, and often insecure. This project addresses these issues by providing a simple yet functional system for user registration and login.

## **Proposed Solution**

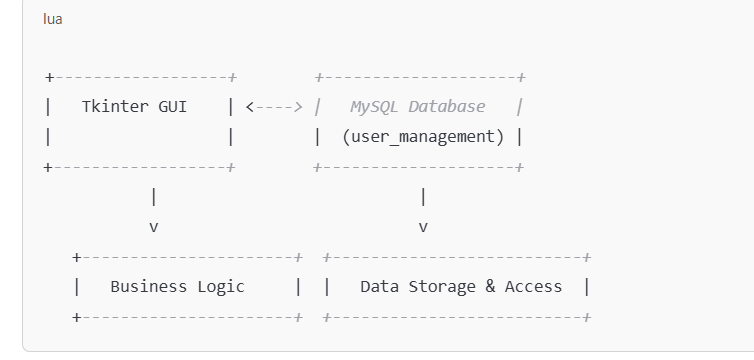
The proposed solution is a user management system that consists of a registration and login interface. Users can create an account by providing a username and password, which are stored in a MySQL database. The system also allows users to log in using their credentials, with appropriate feedback on the success or failure of their attempts.

## **System Architecture**

The system architecture consists of three main components:

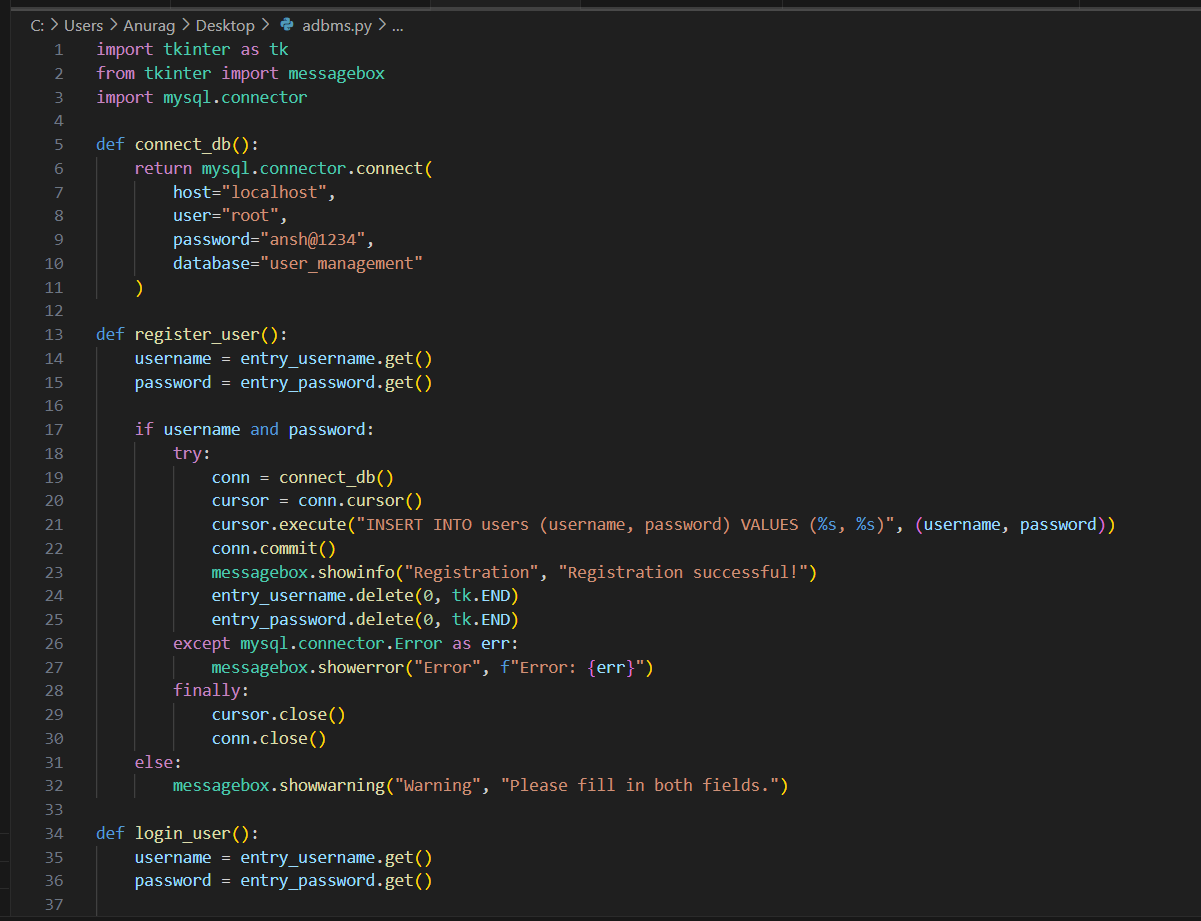
1. **User Interface**: Created using Tkinter, it allows users to interact with the application through forms for registration and login.
2. **Database**: MySQL serves as the backend database to store user credentials securely.
3. **Business Logic**: Python functions handle the registration and login processes, ensuring that user inputs are validated and processed correctly.

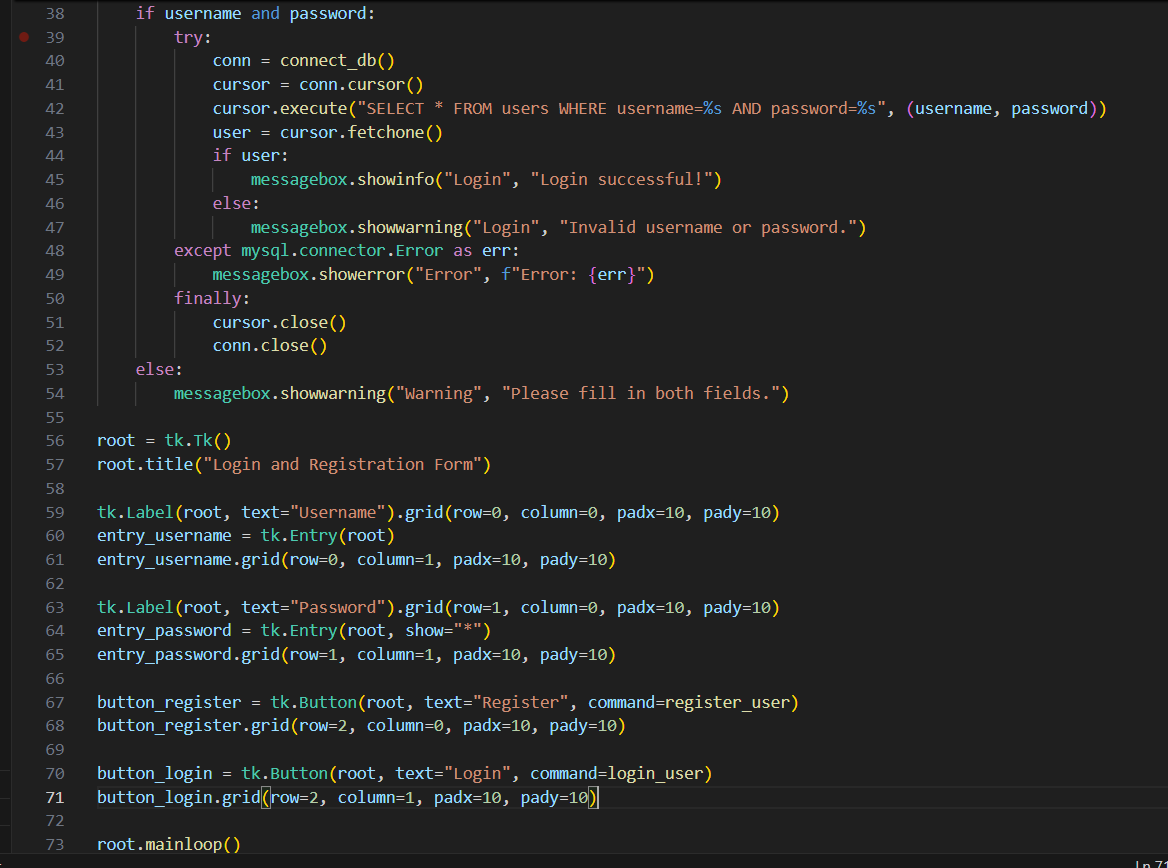
### Diagram

****

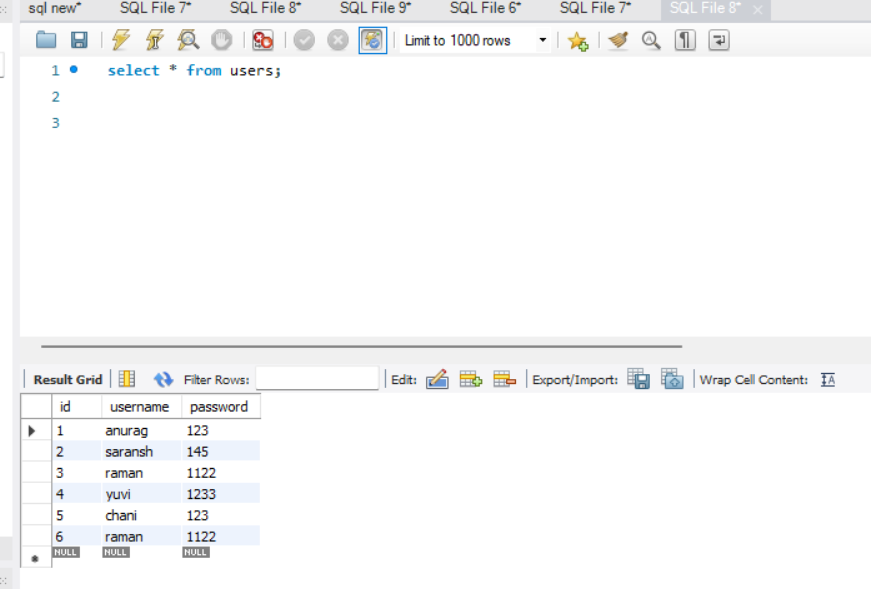
## Implementation

The implementation includes the following Python code that provides the core functionality of the application:

****

****

**Output**

****

## Challenges and Considerations

* **Security**: Storing passwords as plain text poses a significant security risk. Future implementations should incorporate password hashing to protect user data.
* **Error Handling**: The system currently provides basic error messages. More robust error handling could improve user experience by giving detailed feedback.
* **User Input Validation**: Further validation for usernames and passwords (e.g., length, special characters) could enhance security and user experience.

## Future Enhancements

1. **Password Hashing**: Implement password hashing using libraries like bcrypt to store user passwords securely.
2. **User Feedback**: Improve user feedback mechanisms, providing clear guidance on why an action may have failed.
3. **Email Verification**: Introduce email verification during registration to ensure valid user accounts.
4. **User Management Features**: Add features like password recovery, user roles, and user activity logging.

## Conclusion

The user management system created using Tkinter and MySQL demonstrates a basic implementation of user authentication. While it serves its purpose as a simple registration and login system, future enhancements can significantly improve security and usability. This project lays the foundation for more complex applications requiring user management.

## References

* [Python MySQL Connector Documentation](https://dev.mysql.com/doc/connector-python/en/)
* [Tkinter Documentation](https://docs.python.org/3/library/tkinter.html)
* [bcrypt Library Documentation](https://pypi.org/project/bcrypt/)