Python Assignment Report

Kanchi Ram Preetham, 22CS10035

Introduction:

This project is for user registration, authentication, and profile management for teachers, undergraduate (UG) students, and postgraduate (PG) students separately. This report is an overview of the system's features, implementation approach, and user interactions.

1. User Registration:

Registration Process:

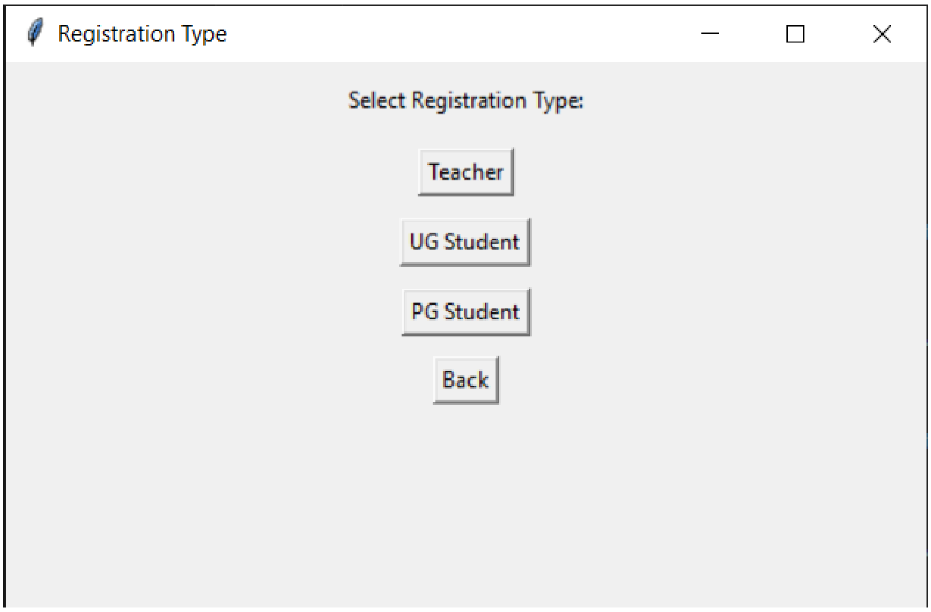
The registration process involves the creation of a new user account. Users are required to provide essential details, including their name, email address (used as a unique Username), and a secure password. The password criteria are followed as given in problem statement:

- Password Strength: Users must create a password within the range of 8-12 characters, including at least one uppercase letter, one digit, and one lowercase letter. Additionally, the password should contain one or more special characters from the set [! @ # $ % & *] . Blank spaces are not allowed.

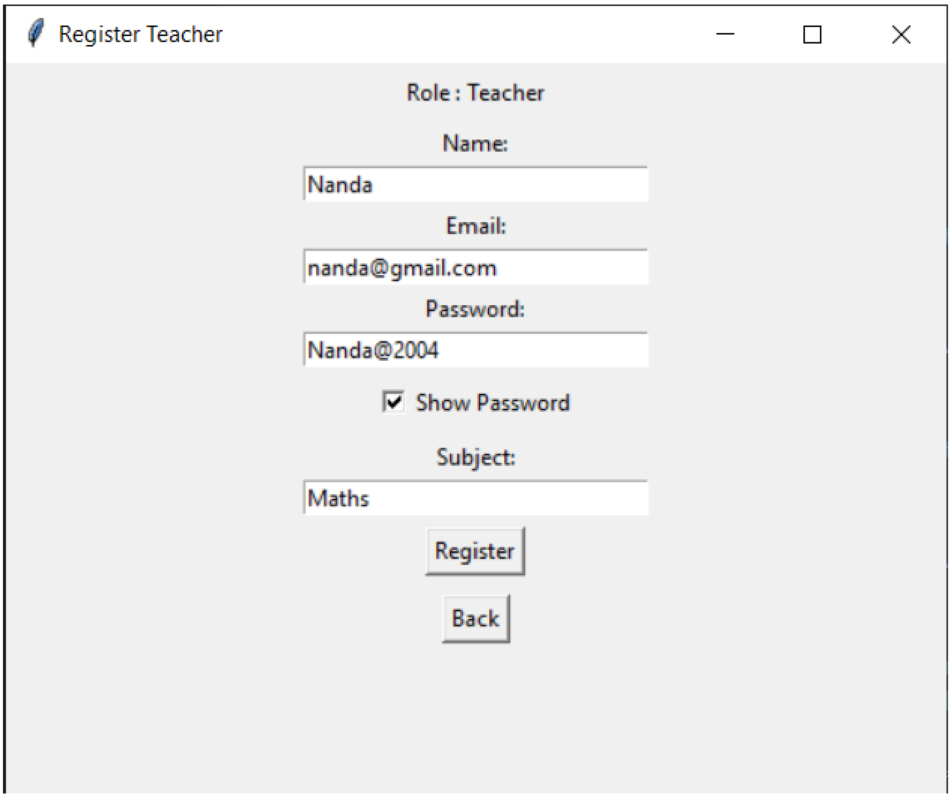
Different User Types:

The system has 3 different user types: teachers, UG students, and PG students. During registration, users select their role, and the registration form dynamically adjusts to collect role-specific information:

- Teacher Registration: In addition to standard details, teachers are required to provide their subject.
- UG Student Registration: UG students need to input their roll number and department.
- PG Student Registration: PG students specify their roll number and specialisation.



Registration type page asking user for type of registration



Register Teacher

Role : Teacher

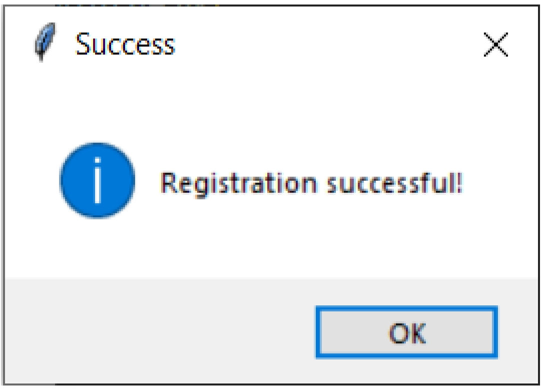
Name: Nanda

Email: nanda@gmail.com

Password: Nanda@2004

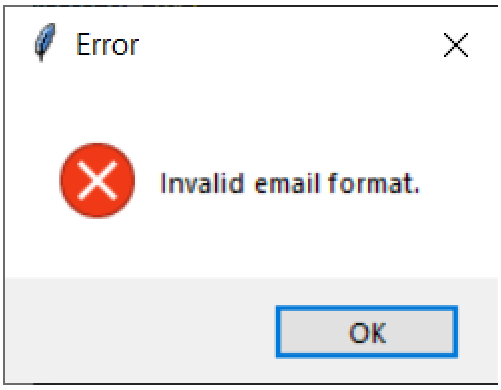
☑ Show Password

Subject: Maths



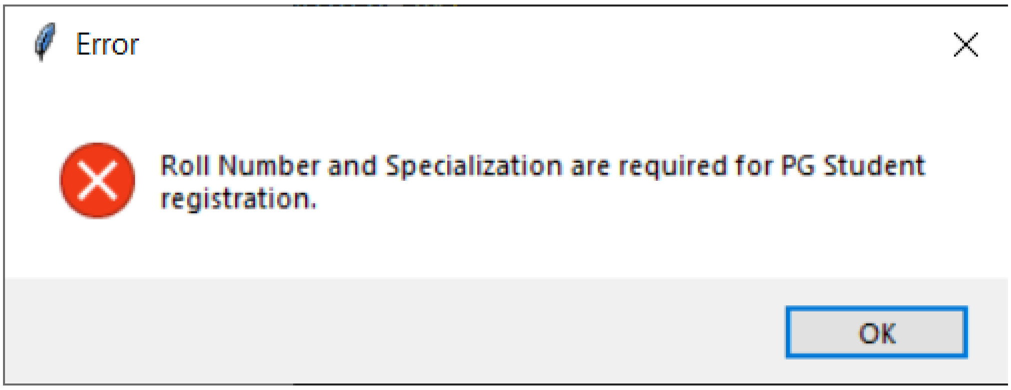
Success

Registration successful!



Error

Invalid email format.



Error

Roll Number and Specialization are required for PG Student registration.

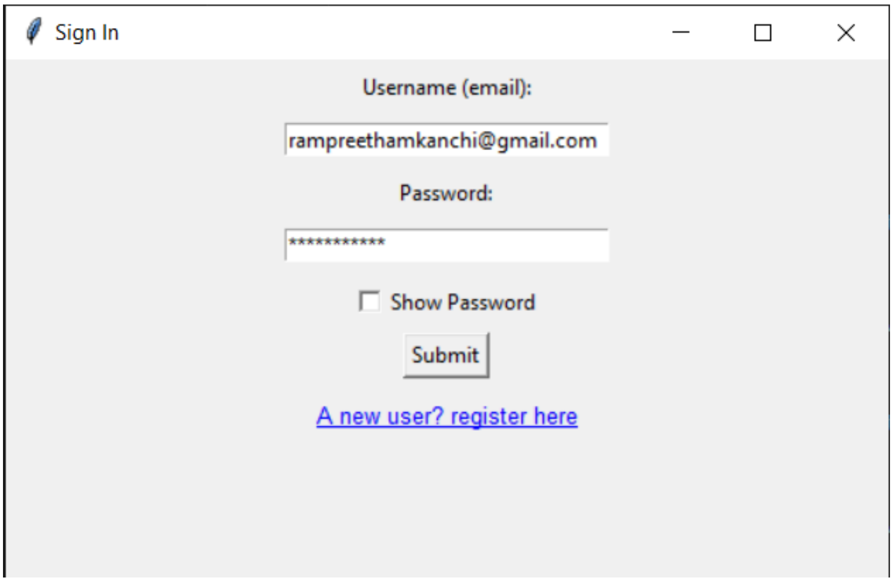
These are screenshots when invalid fields are entered. Registration is successful if valid credentials are entered.

2. Sign-in Process:

Authentication and Security:

The sign-in process involves verifying user credentials against the stored information. Security measures are implemented to prevent unauthorised access:

- Maximum Attempts: Users are allowed three attempts for verification.
- Account Deactivation: After three consecutive incorrect attempts, the account is deactivated.

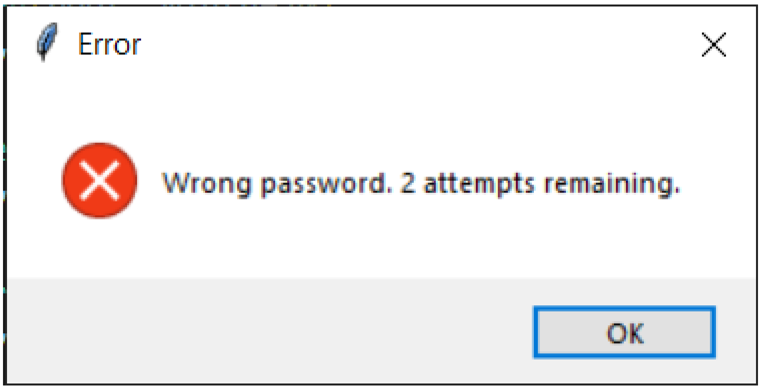


Username (email):

rampreethamkanchi@gmail.com

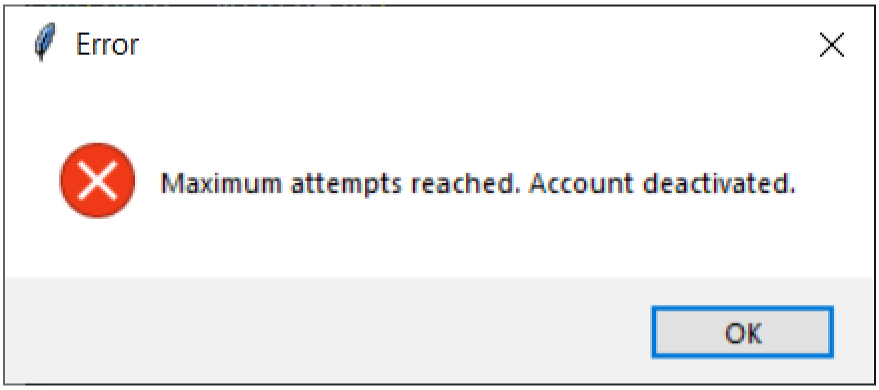
Password:

****



Wrong password. 2 attempts remaining.

Number of remaining attempts are shown after entering wrong password



Error

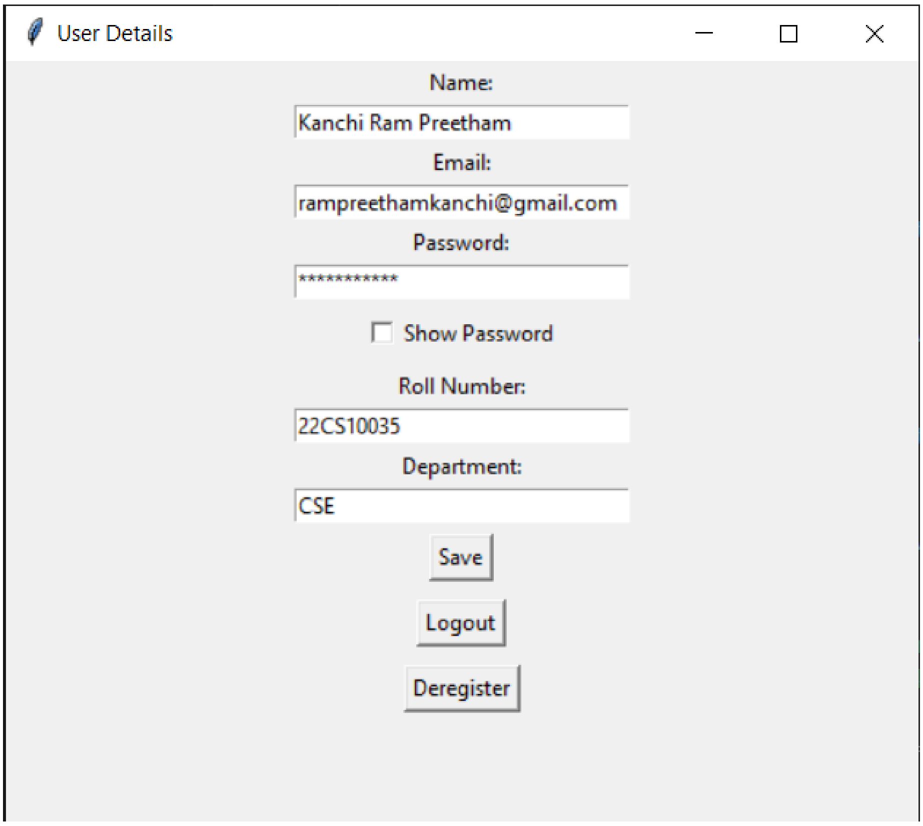
Maximum attempts reached. Account deactivated.

After three consecutive incorrect attempts, the account is deactivated.

3. Profile Management:

User Data Editing:

Once authenticated, users will be shown their profiles for editing. The system allows users to update their information while maintaining the rules of fields.



User Details

Name: Kanchi Ram Preetham

Email: rampreethamkanchi@gmail.com

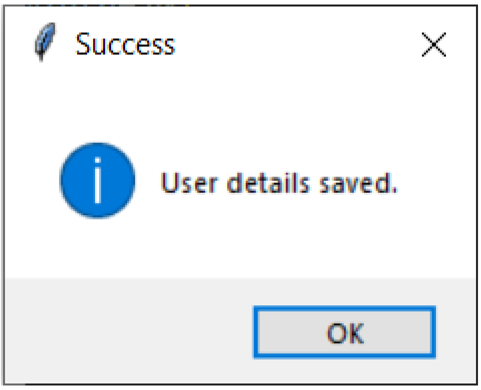
Password: ************

□ Show Password

Roll Number: 22CS10035

Department: CSE

Again checking is done whether proper valid credentials are entered before saving.


Success

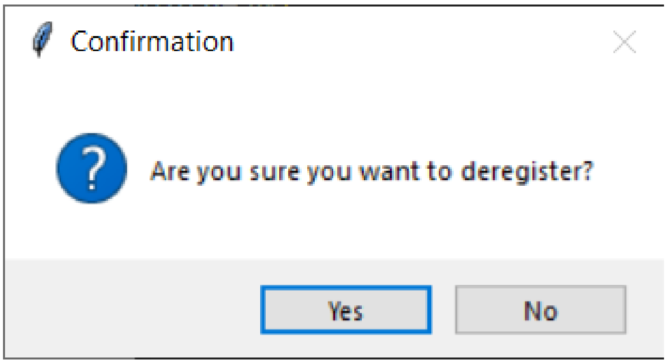
User details saved.

Success message is shown if valid credentials are entered and our data gets updated .

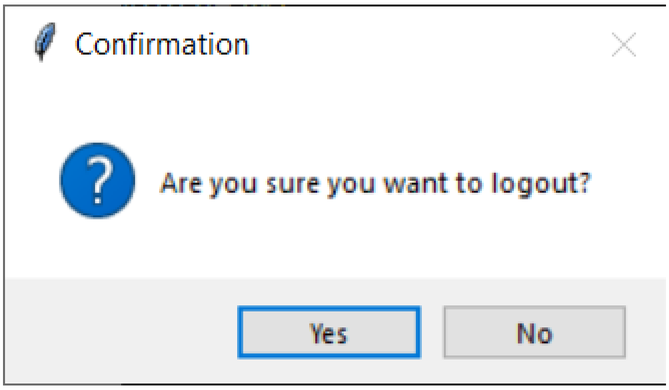
4. Deregistration:

Account Deactivation:

Users can submit deregistration requests when opting to deactivate their accounts. Upon submission, the account is deactivated.



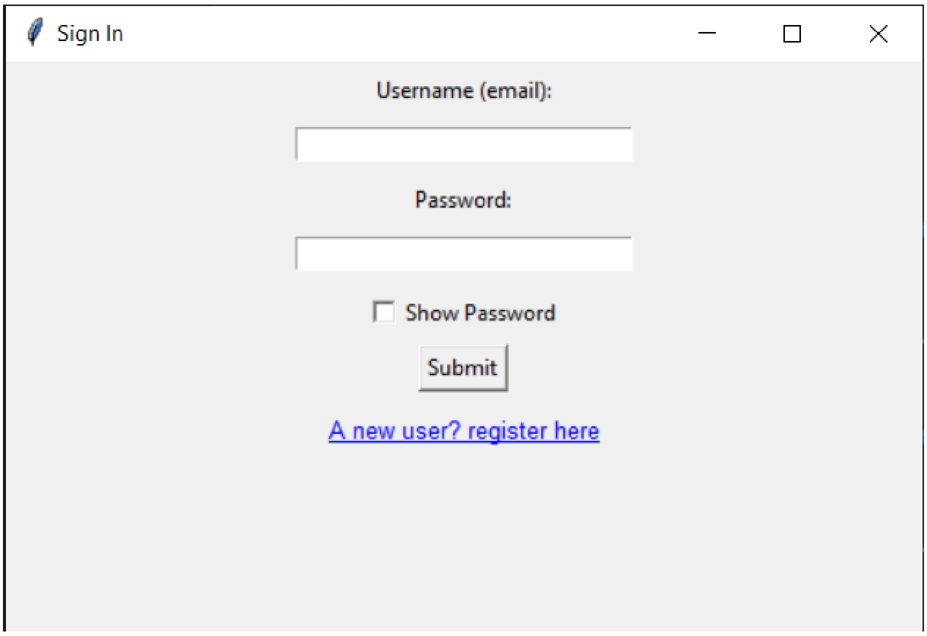
Are you sure you want to deregister?



Are you sure you want to logout?

Before logging out or deactivating the account, a confirmation message is displayed.

If confirmed, data will be updated if necessary and you will be taken back to the sign in page.



Data Management Using Pickle

What is Pickle?

Pickle is a Python module that provides a way to serialise and deserialize Python objects. Serialisation is the process of converting a Python object into a byte stream, and deserialization is the reverse process, where a byte stream is converted back into a Python object. The `pickle` module facilitates the storage and retrieval of objects.

Loading Data

When the academic application starts, it checks for the existence of a file (`users_data.pkl`) containing serialised data. If the file exists and is not empty, the application uses the `pickle.load()` method to deserialize the data and populate the application's data structure (`users_data`). This allows the application to resume with the previously saved user data.

Data Addition

During user registration, the application collects user details based on their role (Teacher, UG Student, PG Student). The entered information is validated, and a new user object is created accordingly. This new user object is then appended to the `users_data` list using the .append() method and saved.

Data Deletion

In response to a deregistration request, the application deletes a user by removing their corresponding user object from the `users_data` list using .remove() method. The updated data is then saved to the pickle file, reflecting the removal of the user from the system.

Saving Data

When the application state is modified, such as during user registration or profile updates, the `pickle.dump()` method is used to serialise the updated data and save it back to the file. This ensures that the changes persist across application sessions.

Extra features

1. Added show password checkbox for toggling from * representation.

2. Implemented logout functionality to go back to sign in page

3. System checks whether a user of the same username (email) exists before and does not allow duplication.