Week 2

Hosain Masba Tamim

Objectives & Outcomes

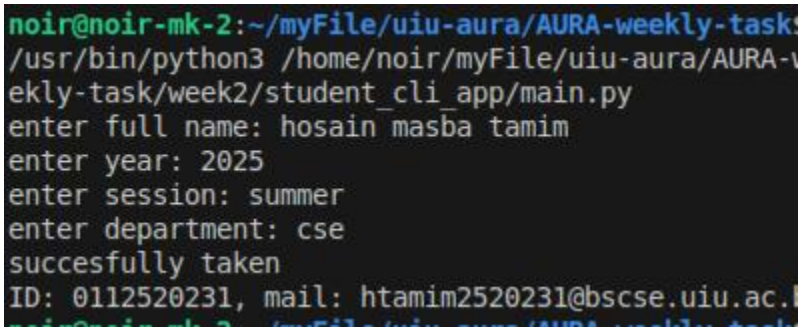
- **Planned**:
  - Learn modular programming and error handling.
- **Achieved**:
  - Basic knowledge about functions and modular programming.
  - Error handling and user defined error handling

CLI App: ***student_cli_app***

Code & Explanations

- **What it does :** This cli app can generate ID and mail of a university student. Users need to input the students name, his/her admission year and session, as well as their department. The app will generate student ID and mail.
- **How it works :** The programme takes input of the students full name, their admission year and session, and department name. The ID is generated by concatenating department ID, batch number and a unique identifier (this unique identifier needs to be manually entered) . The system extracts the last name of the student and the first alphabet of the full name and generates a name key word, and it is then paired with their ID (with out department id of the student id)
- **Key functions :** As this is a cli app, this runs completely on the terminal. No dependency is needed. Users can increase department numbers. Also they can adjust if the student is from trimester or semester. This app also handles all kind of error handling.

- **Screenshots :**



Script: ***input.py***

- **What it does :** This file contains a helper function that collects and validates user details like their full name, admission year, session and department. It makes sure that the entered value is of right data type and handles invalid entries with proper error messages and re-prompts the user until valid data is provided. After successful entry it extracts the last name and first alphabet of the full name, confirms successful input, and returns these values along with the admission year, session, and department.

Script: ***errors.py***

- **What it does :** This file contains user-defined error classes for specific error handling. The base class is BaseError, it inherits the built-in Exception class. It also contains NotValidSession, and NotValidSession to represent distinct error types for invalid sessions and invalid departments.

Script: ***gen.py***

- **What it does :** This file contains all the core functions related to student data validation and ID/mail generation. It imports modules for input handling and custom errors, then defines dictionaries for session, department, and program mappings. The verify_session( ) and verify_department( ) functions validate user-entered session and department values, raising custom errors for invalid inputs.
  The gen_student_id( ) function generates a unique student ID by combining department code, year, session number, and a unique identifier. Finally, the gen_student_main( ) function creates a standardized student email address using the first letter of the name, last name, part of the ID, and the program's code.

Script: ***main.py***

- **What it does :** This file serves as the main entry point of the program. It imports the necessary modules for input handling, error management, and data generation. The program first collects user details through the take_input( ) function, then uses functions from the gen module to validate the department and session, generate a unique student ID, and create an institutional email address. If any validation errors occur, they are caught and displayed using custom error handling. Lastly, the generated student ID and email are printed as the program's output.