InCollege Project: User Profile Creation

Objective: This week, your team will extend the core functionality by enabling users to create and manage their personal profiles within InCollege. This is a crucial step for users to personalize their presence and eventually connect with others effectively. All program input will continue to be read from a file, all output will be displayed on the screen, and that same output will also be written to a file.

Focus Areas:

1. **Profile Information Capture:**
    - Upon successful login, the system will present an option to "Create/Edit My Profile."
    - When a user chooses to create or edit their profile, they should be prompted to enter the following information:
        - **First Name:** (Required)
        - **Last Name:** (Required)
        - **University/College Attended:** (Required)
        - **Major:** (Required)
        - **Graduation Year:** (Required, must be a valid 4-digit year, e.g., 2025)
        - **About Me (Optional):** A short text description where users can provide more details about themselves.
        - **Experience (Optional, up to 3 entries):** Users can add up to three entries for past work or project experience. Each entry should include:
            - Title (e.g., "Software Intern")
            - Company/Organization (e.g., "Tech Solutions Inc.")
            - Dates (e.g., "Summer 2024" or "Jan 2023 - May 2024")
            - Description (Optional, short details about responsibilities/achievements)
        - **Education (Optional, up to 3 entries):** Users can add up to three entries for additional educational background. Each entry should include:
            - Degree (e.g., "Master of Science")
            - University/College (e.g., "State University")
            - Years Attended (e.g., "2023-2025")
2. **Profile Persistence:**
    - All profile information entered by the user must be saved and associated with their respective user account from Week 1.
    - This data should persist across application restarts. You should update your data storage mechanism (e.g., extending the sequential file or creating a new one linked by username) to accommodate this new profile data.
3. **Profile Viewing:**
    - Users should be able to view their own complete profile information once it has been created or updated.
    - Provide an option from the main menu (post-login) to "View My Profile."

4. **Input Validation:**
    - Implement robust validation for all required fields (e.g., ensuring graduation year is numeric and within a reasonable range).
    - Consider handling cases where optional fields are left blank.
5. **I/O Requirements:**
    - **Input:** All user input (e.g., profile details, menu selections) will be read from a predefined input file.
    - **Output Display:** All program output (e.g., prompts, confirmation messages, displayed profile data) must be displayed on the screen (standard output).
    - **Output Preservation:** The exact same output displayed on the screen must also be written to a separate output file for testing and record-keeping purposes.

COBOL Implementation Details (For Programmers):

- **Modular Design:** Create new COBOL modules or sections specifically for handling profile creation, updating, and viewing.
- **Input File Handling:** Continue to implement COBOL `READ` statements to read user input from the designated input file for all profile-related prompts.
- **Output File Handling:** Ensure all program output, including prompts for profile input and the display of profile data, is written to your dedicated output file *identically* to what is displayed on the screen.
- **Data Structures:** Define new COBOL data structures (e.g., `OCCURS` clauses for multiple experience/education entries, `PIC` clauses for various data types) to store the detailed profile information for each user.
- **Persistence File I/O:** Extend your file handling logic from Week 1 to store and retrieve the new profile data. Consider how to link a user's profile to their login credentials (e.g., using the username as a key).
- **User Interaction:** Design clear console prompts for users to input their profile details, keeping in mind these prompts will be echoed to your output file.

Testing Responsibilities (For Testers):

- **Test Case Development:** Create comprehensive test cases in Jira for all Week 2 functionalities, including:
    - **Positive Test Cases:** Scenarios for successful profile creation/editing with all required fields, and viewing a complete profile. Include scenarios with optional fields both present and omitted.
    - **Negative Test Cases:** Scenarios for invalid input (e.g., non-numeric graduation year, invalid year range, blank required fields).
    - **Edge Cases:** Test boundaries, like exactly 3 experience/education entries, or omitting all optional fields.
- **Test Execution:** Execute all developed test cases using the specified input file.
- **Bug Reporting:** For every issue or discrepancy found, create a detailed bug ticket in Jira. Include steps to reproduce, actual results, and expected results.

- **Output Verification:** Meticulously compare the program's console output against the generated output file to ensure they are absolutely identical for all profile creation, editing, and viewing scenarios.
- **Collaboration:** Work closely with the programmers to help them understand and reproduce bugs.

Jira Requirements (For Scrum Master, Programmers, & Testers):

Your team's Jira board for Week 2 should include:

- **Epic #2: User Profile Management:** Define this new epic.
- **User Stories:**
  - "As a logged-in user, I want to create my personal profile so others can learn about my background."
  - "As a logged-in user, I want to edit my personal profile so I can keep my information up-to-date."
  - "As a logged-in user, I want to view my personal profile so I can see how it appears to others."
  - "As a user creating a profile, I want to enter my name, university, major, and graduation year."
  - "As a user creating a profile, I want to add an 'About Me' section."
  - "As a user creating a profile, I want to add multiple work experience entries."
  - "As a user creating a profile, I want to add multiple education entries."
  - "As a user, I want my profile information to be saved permanently."
  - **Updated User Story (for Testing):** "As a tester, I want the program to read all user inputs for profile creation from a file so I can automate testing."
  - **Updated User Story (for Testing):** "As a tester, I want the program to write all screen output related to profile management to a file so I can easily verify results."
- **Tasks:** Break down each User Story into granular tasks that individual team members can work on. (e.g., *Programmers:* "Define COBOL data structures for profile fields," "Implement COBOL module for profile data write/read," "Add input validation for graduation year," "Integrate profile creation option into main menu," "Update input/output routines for profile management to use file I/O." *Testers:* "Develop test cases for profile creation/editing," "Execute profile viewing tests," "Log bugs related to profile management," "Verify I/O consistency for profile features").
- **Bug Tickets:** Log any issues found during development and testing.

GitHub Requirements:

- **New Modules/Files:** Commit your new COBOL modules or updated existing files for profile management.
- **Branching:** Continue to use your team's established branching strategy for new features.

- **Regular Commits:** Ensure programmers are making frequent, meaningful commits with clear messages. Testers should commit their test files.
- **README.md Update:** Update your README.md to reflect the new profile creation and viewing capabilities, explicitly noting how to prepare input for profile management and where to find the corresponding output.

Deliverables for End of Week 2:

1. **Roles.txt**: List of team members and the roles that they played this week.
2. **InCollege.cob: Working COBOL Program:** A console-based COBOL application that allows users to create/edit their profiles, saves this information persistently, and allows users to view their own profiles. This should seamlessly integrate with the login functionality from Week 1. All inputs must be read from a file, all outputs displayed on the screen, and the exact same outputs written to a separate file.
3. **InCollege-Input.txt: Sample Input File:** A sample text file demonstrating the format of input your program expects for Week 2's functionality.
4. **InCollege-Output.txt:** A sample text file showing the expected output for a typical run of your program, demonstrating profile creation/editing and viewing.

```
--- SAMPLE_OUTPUT_WEEK2.TXT ---
Welcome to InCollege!
1. Log In
2. Create New Account
Enter your choice:
Please enter your username:
Please enter your password:
You have successfully logged in.
Welcome, TestUser!
1. Create/Edit My Profile
2. View My Profile
3. Search for User
4. Learn a New Skill
Enter your choice:
--- Create/Edit Profile ---
Enter First Name:
Enter Last Name:
Enter University/College Attended:
Enter Major:
Enter Graduation Year (YYYY):
Enter About Me (optional, max 200 chars, enter blank line to skip):
Add Experience (optional, max 3 entries. Enter 'DONE' to finish):
Experience #1 - Title:
Experience #1 - Company/Organization:
Experience #1 - Dates (e.g., Summer 2024):
Experience #1 - Description (optional, max 100 chars, blank to skip):
Add Experience (optional, max 3 entries. Enter 'DONE' to finish):
DONE
Add Education (optional, max 3 entries. Enter 'DONE' to finish):
Education #1 - Degree:
```

```
Education #1 - University/College:
Education #1 - Years Attended (e.g., 2023-2025):
Add Education (optional, max 3 entries. Enter 'DONE' to finish):
DONE
Profile saved successfully!
1. Create/Edit My Profile
2. View My Profile
3. Search for User
4. Learn a New Skill
Enter your choice:
--- Your Profile ---
Name: John Doe
University: Example University
Major: Computer Science
Graduation Year: 2025
About Me: Enthusiastic developer always learning new things.
Experience:
  Title: Software Intern
  Company: Tech Corp
  Dates: Summer 2024
  Description: Developed tools for internal testing.
Education:
  Degree: Bachelor of Science
  University: Example University
  Years: 2021-2025
--------------------
1. Create/Edit My Profile
2. View My Profile
3. Search for User
4. Learn a New Skill
Enter your choice:
--- END_OF_PROGRAM_EXECUTION ---
```

5. **Epic2-Storyx-Test-Input.zip: Test Input Files:** A set of test input files used by the testers, covering positive, negative, and edge cases for each of this week's stories.
6. **Epic2-Storyx-Test-Output.zip: Actual Test Output Files:** The exact output generated by running your program with the Epic2-Storyx-Input input Files, submitted for review.
7. **Jira.jpg: Updated Jira Board:** All relevant User Stories, tasks, and bugs (with their status) for Week 2 should be updated in Jira.
8. **GitHub.jpg:** Go to the repository's main page. Click the "Commits" link (next to the green "Code" button). Show a chronological list of all commits with messages, authors, and timestamps

Your testers will have a critical role this week in ensuring all profile fields are correctly saved and retrieved, validating input, and verifying that changes to the profile are accurately reflected. They will also pay close attention to the correctness and consistency of the file-based I/O. The scrum master will continue to facilitate the team's progress and ensure any blockers are addressed promptly.