

InCollege Project: Week 7 Deliverable - Browse & Applying to Job/Internships (with Report Enhancement)

Objective: This week, your team will complete the basic job board by enabling users to browse all available job and internship listings, simulate the application process, and *generate a summary report of their own applications*. 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. Job/Internship Browse:

- The "Browse Jobs/Internships" option, which was previously an "under construction" message, will now become fully functional.
- When a user selects this option, the system should display a list of *all* available job and internship postings that have been created.
- Each listing should show at least the Job Title, Employer, and Location to provide a quick overview.

2. Viewing Full Job Details:

- Users should be able to select a specific job listing from the browsed list to view its full details (Job Title, Description, Employer, Location, and Salary if provided).

3. Simulated Application Process:

- When viewing the full details of a job, the system should present an option to "Apply for this Job."
- When a user chooses to apply, the system should simply register their interest. For this alpha version, a complex application process (like uploading resumes or cover letters) is *not* required.
- The system should record that the logged-in user has "applied" to this specific job. This data should persist.
- The user should receive a confirmation message that their application has been submitted (e.g., "Your application for [Job Title] at [Employer] has been submitted.").

4. Job Application Summary Report Generation:

- After successfully applying for a job, or from a new option in the "Job Search/Internship Menu" (e.g., "View My Applications"), the user can request a report.
- This report will list all job/internship postings that the currently logged-in user has applied for.
- For each application, the report should display:
 - The Job Title
 - The Employer
 - The Location
 - (Optional but good) A unique identifier for the application or the job.
- The report should include a simple header (e.g., "Your Job Applications") and potentially a count of total applications.

- This report output must adhere to the standard I/O requirements: displayed on screen and written to the output file.
- 5. **I/O Requirements:**
 - **Input:** All user input (e.g., menu selections, job selections, application confirmations, report request) will be read from a predefined input file.
 - **Output Display:** All program output (e.g., prompts, confirmation messages, displayed job lists, full job details, and the new report) 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):

- **Job Listing Retrieval & Display:** Develop COBOL modules to read all job postings from your persistent storage (created in Week 6) and display them in a list format.
- **Detailed Job Display:** Implement COBOL routines to retrieve and display the full details of a selected job posting.
- **Application Data Structure:** Define a new COBOL data structure (e.g., a new sequential file or an addition to an existing one) to store job applications. This structure will need fields to link the applying user (their username) with the specific job posting (e.g., using a job ID or a combination of job details).
- **File I/O for Applications:** Implement COBOL file operations to write new application records to your persistent storage.
- **Report Generation Module:** Develop a dedicated COBOL module or section for generating the "Job Application Summary Report." This will involve:
 - Reading the persistent application data.
 - Filtering records to only include applications by the logged-in user.
 - Formatting the output for the report header, detail lines, and summary (e.g., total count).
 - Using DISPLAY for screen output and WRITE for the output file simultaneously for the report.
- **Input File Handling:** Continue to implement COBOL READ statements to read user input from the designated input file for all menu selections, job selections, application confirmations, and report requests.
- **Output File Handling:** Ensure all program output, including prompts, displayed job information, and the generated report, is written to your dedicated output file *identically* to what is displayed on the screen.
- **Menu Integration:** Ensure the "Browse Jobs/Internships" option navigates correctly, the "Apply for this Job" option appears, and *add the new "View My Applications" option* within the "Job search/internship" sub-menu.

Testing Responsibilities (For Testers):

- **Test Case Development:** Create comprehensive test cases in Jira for all Week 7 functionalities, including:
 - **Browse Jobs:** Scenarios for viewing an empty job list, a list with one job, and a list with multiple jobs.
 - **Viewing Job Details:** Test selecting existing jobs and attempting to select non-existent job numbers.
 - **Applying to Jobs:** Scenarios for successfully applying to a job.
 - **Report Generation:** Test generating the report when the user has no applications, one application, and multiple applications. Verify report formatting (header, detail lines, total count).
 - **Persistence:** Ensure job applications and the data used for the report persist across program restarts.
- **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 job Browse, application, and report generation 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 7 should continue building on **Epic #4: Job Board Functionality** and include:

- **User Stories for Browse Jobs:**
 - "As a logged-in user, I want to browse all available job and internship listings."
 - "As a user Browse jobs, I want to see a summary of each listing (title, employer, location)."
 - "As a user, I want to view the full details of a specific job posting once I select it."
- **User Stories for Applying to Jobs:**
 - "As a logged-in user, I want to apply for a job or internship posting."
 - "As a user applying for a job, I want my application to be recorded persistently."
 - "As a user, I want to receive a confirmation message after applying to a job."
- **NEW User Stories for Application Report:**
 - "As a logged-in user, I want to view a report summarizing all the jobs/internships I have applied for."
 - "As a user viewing the application report, I want to see the job title, employer, and location for each applied position."
 - "As a user viewing the application report, I want a clear header and a count of my total applications."

- **New User Story (for Testing):** "As a tester, I want the program to read all user inputs for job Browse, application, *and report generation* from a file so I can automate testing."
- **New User Story (for Testing):** "As a tester, I want the program to write all screen output related to job Browse, application, *and report generation* 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:* "Develop COBOL routine to read and display all job summaries," "Implement COBOL module to display full job details," "Define COBOL file structure for job applications," "Create COBOL routine to record an application," "**Develop COBOL module for application report generation,**" "**Integrate 'View My Applications' option into menu,**" "Update I/O for job Browse, application, and report generation features to use file input/output consistently." *Testers:* "Develop test cases for job Browse/details," "Develop test cases for applying to jobs," "Develop test cases for application report generation," "Execute all Week 7 tests," "Log bugs," "Verify I/O consistency").
- **Bug Tickets:** Log any issues found during development and testing.

GitHub Requirements:

- **New Modules/Files:** Commit any new COBOL modules or updated existing files for Browse jobs, handling applications, *and generating the application report*.
- **Branching:** Continue to follow your team's established branching strategy for new features.
- **Regular Commits:** Ensure consistent, descriptive commits throughout the week. Testers should commit their test files.
- **README.md Update:** Update your README.md to reflect the new functionality, explicitly detailing how to prepare input for these features and where to find the corresponding output, including the new report.

Deliverables for End of Week 7:

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 logged-in users to:
 - Navigate to the "Job search/internship" menu.
 - Select an option to "Browse Jobs/Internships" and see all available listings.
 - Select a listing to view its full details.
 - "Apply" for a job, with the application being recorded persistently and a confirmation message displayed.
 - Generate and display a "Job Application Summary Report" of their own applications.

- This must seamlessly integrate with all previous weeks' functionality. 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 7's functionality (e.g., menu choices, job selection ID, application confirmation, report request).
- 4. **InCollege-Output.txt:** A sample text file showing the expected output for a typical run of your program, demonstrating Browse, applying to jobs, *and generating the application report*.

```

--- SAMPLE_OUTPUT_WEEK7_WITH_REPORT.TXT ---
Welcome to InCollege!
    Log In
    Create New Account
Enter your choice:
Please enter your username:
Please enter your password:
You have successfully logged in.
Welcome, TestUser!
    Search for a job
    Find someone you know
    Learn a new skill
    View My Pending Connection Requests
    View My Network
Enter your choice:
--- Job Search/Internship Menu ---
    Post a Job/Internship
    Browse Jobs/Internships
    View My Applications
    Back to Main Menu
Enter your choice:
--- Available Job Listings ---
    Software Intern at TechCorp (New York, NY)
    Data Analyst at DataSolutions (Remote)
    Marketing Specialist at BrandInc (Chicago, IL)
-----
Enter job number to view details, or 0 to go back:
--- Job Details ---
Title: Software Intern
Description: Develop tools for internal testing and automation.
Employer: TechCorp
Location: New York, NY
Salary: $25/hour
-----
    Apply for this Job
    Back to Job List
Enter your choice:
Your application for Software Intern at TechCorp has been submitted.
--- Available Job Listings ---
    Software Intern at TechCorp (New York, NY)
    Data Analyst at DataSolutions (Remote)

```

```

Marketing Specialist at BrandInc (Chicago, IL)
-----
Enter job number to view details, or 0 to go back:
0
--- Job Search/Internship Menu ---
        Post a Job/Internship
        Browse Jobs/Internships
        View My Applications
        Back to Main Menu

Enter your choice:
--- Your Job Applications ---
Application Summary for TestUser
-----
Job Title: Software Intern
Employer: TechCorp
Location: New York, NY
---
Job Title: Data Analyst
Employer: DataSolutions
Location: Remote
-----
Total Applications: 2
-----
--- Job Search/Internship Menu ---
        Post a Job/Internship
        Browse Jobs/Internships
        View My Applications
        Back to Main Menu

Enter your choice:
--- END_OF_PROGRAM_EXECUTION ---

```

5. **Epic-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. **Epic7-Storyx-Test-Output.zip: Actual Test Output Files:** The exact output generated by running your program with the Epic7-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 3 should be updated in Jira.
8. **Jira:** Two Burndown charts. The first created on Monday and the second created when the Sprint is complete.
9. **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 be critical this week in verifying that all job listings are displayed correctly, that the full details appear when a job is selected, that the application process successfully records the user's interest, *and that the new application report accurately reflects the user's applications*. They will also meticulously compare the console output with the generated output file to ensure perfect consistency across all features, including the report. The scrum master will continue to facilitate daily stand-ups and remove any impediments for the team.

