**S428 Spring 2025 Final Project: Campus BookShare (or your website)**

**Part B – Validation, Testing, and Deployment**

**Objective:**

In Part B, your team will enhance your full-stack web application from Part A—whether you chose the Campus BookShare example or brought your own idea—by adding robust validation, comprehensive testing, and deployment readiness. These additions are designed to simulate real-world development practices and improve the quality, usability, and shareability of your project.

1. **Enhanced Server-Side Validation**

Your application must include meaningful server-side validation for all forms that handle user input.

As as an example for the BookShare application this might include:

* Resource listings (e.g., required fields, valid ISBN format, availability values)
* User profile updates (e.g., proper email format, location not blank)
* Review and message submissions (e.g., rating must be numeric and between 1–5)
* Registration and login (e.g., strong passwords, existing email check)

Although not covered in class, you may use Flask-WTF or implement custom logic in your route handlers. Clearly demonstrate these validations with error handling and feedback to users.

2. **Test Plan and Implementation**

**A. Test Plan Document:** Create a test plan (MS Word or PDF) outlining:

* What features will be tested
* Types of testing used (manual unit testing, system testing, user acceptance testing)
* Testing method (manual walkthroughs, checklists, screenshots, expected vs. actual results)
* Example test cases (inputs, expected outcomes)

**B. Manual Testing Execution:** As part of your submission, include documentation of your team’s manual testing process. This should include:

* At least 5 unit-level manual tests: Confirm that individual functions, forms, or components work correctly in isolation.
* At least 3 system-level tests: Walk through major workflows (e.g., listing a book, reserving it, receiving a notification).
* At least 2 user acceptance tests: Simulate end-user scenarios to confirm the app meets real-world needs (e.g., student browsing listings and successfully making a reservation).
* Provide screenshots or annotated steps showing test execution and outcomes.
* Include a column in your test plan for “Pass/Fail” status and tester initials or name.

3. **GitHub Repository & Instructor Access**

Ensure your GitHub repository is updated, clean, and ready for instructor review.

* Confirm the repo includes all code, assets, and documentation
* Add your instructor (GitHub username will be provided) as a **collaborator** with read access
* Push all code to main or main/dev branches as appropriate
* Create or update your README.md with:
  + Overview of the project and team
  + Setup instructions
  + Deployment link (if applicable)
  + Test plan and how to run tests
* Submit a link to the repository in the Canvas submission portal

Part B Deliverables

* Updated application with working server-side validation
* Test plan document
* Manual test evidence (completed test plan, screenshots, notes, or user testing artifacts)
* Updated GitHub repository with instructor access and full documentation

Evaluation Criteria (Part B Only - 30 Points)

|  |  |
| --- | --- |
| Category | Points |
| Server-Side Validation | 10 |
| Test Plan Quality | 5 |
| Manual Test Execution (unit, system, user acceptance) | 10 |
| GitHub Sharing & Documentation | 5 |

**Reminder:** Part B builds on your Part A project. Make sure all core features from Part A remain functional while adding new capabilities required in Part B. This applies to both the provided Campus BookShare example and custom project ideas.

I'll create a focused test plan specifically for the form validations and error handling we implemented.

# Test Plan: Form Validation and Error Handling

**1. Features to be Tested**

- User Registration Form Validation

- User Login Form Validation

- Blog Post Creation Form Validation

- Edit Post Validation

**2. Testing Methods**

- Manual form submission testing

- Error message documentation

**3. Test Cases**

**Registration Test Cases**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Registration Base Page:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Registration Test Results:**

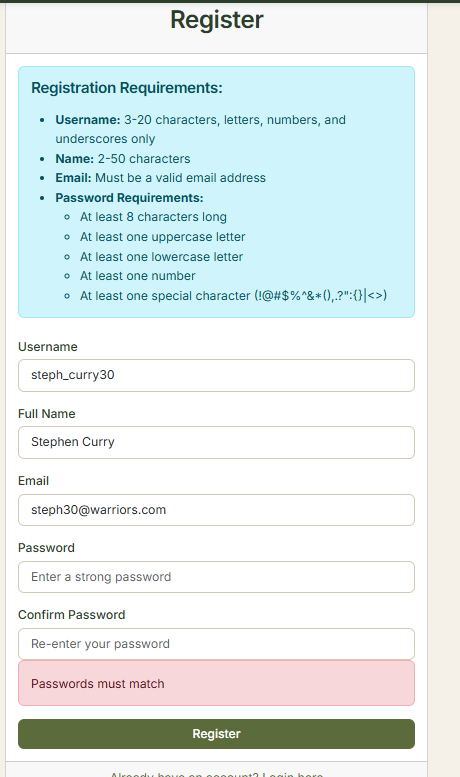
**1. A screenshot of a computer

AI-generated content may be incorrect.**

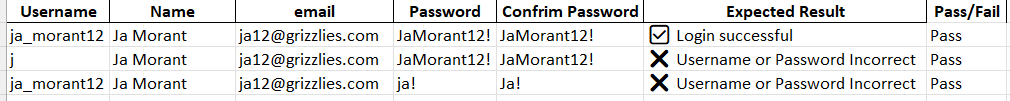
**2. A screenshot of a computer

AI-generated content may be incorrect.3.** **A screenshot of a computer

AI-generated content may be incorrect.4.** **A screenshot of a computer screen

AI-generated content may be incorrect.5.** ****

**Login Test Cases**

****

**Login Test Results:**

**1. A screenshot of a computer

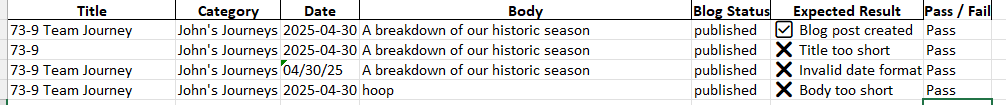
AI-generated content may be incorrect.**

**2. A screenshot of a login screen

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.**

**Create Blog Post Test Cases**

****

**Create Blog Post Test Results:**

1. **A screenshot of a computer

   AI-generated content may be incorrect.A screenshot of a computer

   AI-generated content may be incorrect.2. A screenshot of a computer

   AI-generated content may be incorrect.3. A screenshot of a computer

   AI-generated content may be incorrect. 4. A screenshot of a computer

   AI-generated content may be incorrect.**

**Edit Blog Test Cases**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Edit Blog Test Results:**

1. **A screenshot of a computer

   AI-generated content may be incorrect.2.** **A screenshot of a computer

   AI-generated content may be incorrect.3. A screenshot of a computer

   AI-generated content may be incorrect.4. A screenshot of a computer

   AI-generated content may be incorrect.**