Bynry QA Internship Assignment
Submitted by: Vishal Dwivedy
Date: May 29, 2025

Day 1 Package

1. Critical Observations

Bug – "Download Receipt" & "Update Account Info" Buttons Do Nothing

Clicking these buttons results in no action, no error, and no feedback. For a billing platform, this is critical: users expect to download receipts for proof and update personal info with confidence. A silent failure damages trust and usability.

Risk – No Payment Confirmation / Receipt

After submitting a payment, there's no clear confirmation or downloadable receipt. This is a major risk for a billing platform — customers need proof of payment for peace of mind and dispute resolution.

Feature That Works Well – Responsive UI

Across mobile and desktop, the layout adapts well. Navigation stays smooth and intuitive. This is a solid foundation to build on — especially important since users may be accessing the service from a wide range of devices.

Recommendation – Improve Feedback for User Actions

There's very little feedback for invalid input or errors (e.g., wrong bill number). Implementing clear, contextual error messages can dramatically reduce user frustration and support tickets.

Strategic Question – No Guest Checkout?

Requiring login before initiating payment may be a conversion killer. For utility services, a guest mode could help users make faster transactions and reduce drop-offs — especially for one-time users. Was this a conscious design tradeoff?

---

2. Elderly User Experience Simulation

Where did I get confused?

- The homepage was visually cluttered with no clear entry point.
- Font size was too small, especially for someone who uses reading glasses.
- Buttons like "Pay Bill" were not visually prominent.
- No confirmation or guidance appeared after entering bill details or clicking payment-related actions.
- The entire process lacked visual cues about progress or what step I was on.

What would make someone give up?

- Not knowing what to click or do next.
- Tiny, low-contrast text that's hard to read.
- Pressing a button and nothing visible happening.
- No confirmation that anything worked.
- Anxiety from dealing with money in an unclear system.

What would help?

- Larger fonts and high-contrast buttons for core actions.
- A "Start Here" button to simplify the user flow.
- Visual progress bar to guide users through steps.
- Clear and gentle error messages (e.g., "Could not find your bill — try checking the number again.")
- Optional "Accessibility Mode" with simplified layout and visuals.

---

3. Developer vs. Customer's Son Explanation

To the Developer (Technical & Actionable):

"The current payment flow is not accessible or intuitive, especially for older users. Key UI elements like labels, buttons, and feedback mechanisms are either too small or missing entirely. There's no progress bar, error messages are vague or absent, and ARIA/accessibility standards are unmet. Recommend: 16px+ font base, WCAG-compliant contrast, ARIA labels, a clear CTA, and contextual error handling."

To the Customer's Son (Simple & Empathetic):

"Thanks for testing it for your mom — you're right to be concerned. The site is confusing, especially for older folks. The text is small, it's hard to tell what to click, and it doesn't really 'talk back' when something goes wrong. A clearer layout, bigger buttons, and gentle guidance would make a big difference. I've passed these ideas to the team. She deserves a smooth, simple experience — and we're working on it."

Day 2 Package

4. Pre-Coding Thought Questions

1. What does "works" mean for a Pay Bill button?

- The button is visible, clickable, and enabled.
- It correctly triggers payment flow (navigates or opens modal).
- Backend confirms request handling (e.g., 200/201 or success response).
- The user sees confirmation, success message, or receipt.

2. How would I test without charging money?

- Use a sandbox environment or test credentials.
- If unavailable, use mock data or intercept network calls.
- Abort the flow before actual payment confirmation.
- Test with $0.01 or invalid consumer number to trigger safe paths.

3. What could go wrong that the script should catch?

- Button not rendering or being disabled.
- HTTP errors (404, 500) after button click.
- Redirect failures or page freezing.
- Missing form fields or confirmation messages.
- Console or network errors.

4. How would I know why it broke?

- Capture JS console logs and network errors.
- Check element visibility and class state.
- Monitor status codes and response messages.
- Log each step so we know where it stopped.

---

5. Test Script

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.chrome.options import Options
import time

# Headless mode
options = Options()
options.add_argument("--headless")
driver = webdriver.Chrome(options=options)

try:
    # STEP 1: Open site
    driver.get("https://example-utility-payment.com")
    time.sleep(2)

    # STEP 2: Login
    driver.find_element(By.ID, "username").send_keys("testuser")
    driver.find_element(By.ID, "password").send_keys("testpass", Keys.RETURN)
    time.sleep(2)

    # STEP 3: Locate and check Pay Bill button
    pay_button = driver.find_element(By.ID, "pay-bill-button")
    if pay_button.is_displayed() and pay_button.is_enabled():
```

```
        print("Pay Bill button is visible and enabled.")

        # STEP 4: Click and observe
        pay_button.click()
        time.sleep(2)

        # STEP 5: Check result
        if "payment" in driver.current_url or "amount" in driver.page_source.lower():
            print("Button click leads to payment flow.")
        else:
            print("Unexpected result. Check redirection or flow integrity.")
    else:
        print("Pay Bill button is not visible or is disabled.")

except Exception as e:
    print("Error during test run:", e)

finally:
    driver.quit()
```

---

6. Business Decision Summary

Recommendation: Delay the Launch.

We found blockers that cause confusion during payments — especially for older or non-technical users. Launching as-is risks failed payments, trust loss, and customer churn. The cost of delay is high, but the cost of user frustration and brand damage is higher. I recommend launching a beta version with limited access, collecting real feedback while we fix these key issues.

---

7. Follow-Up Questions

What would change your decision?

- Proof that critical issues don't affect majority users (e.g., A/B testing data).
- Evidence of fallback mechanisms or live support during launch.
- Agreement on post-launch fixes via feature flags.

How would I monitor a launch?

- Real-time error logging (e.g., Sentry).
- Funnel analytics (drop-offs, failed submissions).
- Session replays or heatmaps.
- Alerts for support ticket spikes.

Backup plan if things fail:

- Instant banner guidance + contact info.
- Rollback to last stable version.
- Issue free credits or refunds.
- Public apology + roadmap update.

Final Thought: The Real Test

The part that worries me most is the absence of clear confirmation at the end of the payment process. My grandmother wouldn’t know if she actually paid or not. That’s a scary feeling when money’s involved. She might pay twice — or not at all — because she can’t trust what she sees. It’s not just about usability. It’s about giving users confidence.

Appendix: Evidence & Process Notes

- **Screenshots Attached**: unclear buttons, broken login, small fonts, missing confirmation messages, layout issues.
- **Process Notes**: test credentials used, error logs from console, page source review, and user simulation paths.

Personal Note to the Bynry Team

I absolutely loved working on this assignment. It challenged my thinking, pushed me to balance empathy and engineering, and reminded me why great QA is about people first. I'm eager to work smart, contribute fast, and help build something that's not just functional — but truly trusted by users of all kinds.