

Test Plan – Todo App QA Challenge

1. Overview

This test plan outlines the strategy for verifying both the frontend (React) and backend (Node.js/Express) of a simple Todo application from github. The goal is to ensure that the application supports basic functionality such as user authentication and todo creation.

2. What Is Being Tested

| Area | Description |
|-------------|--|
| UI | User login, todo creation, basic interactions |
| API | Authentication, todo creation, retrieval |
| Integration | UI reflects backend changes (via CRUD overlap) |

3. Test Coverage

UI Tests (Selenium + Pytest)

- Valid login flow
- Invalid login error handling
- Creating a new todo item
- UI feedback on successful todo creation

API Tests (Postman + Newman)

- POST /login – Valid user credentials
- POST /todos – Add new todo
- GET /todos – Retrieve todos for user
- PUT and DELETE – Not covered: routes not implemented in app

4. Tools Used

| Tool | Purpose | Justification |
|--------------------------|--------------------|-----------------------------------|
| Selenium (Python) | UI testing | Simulates user actions on browser |
| Pytest | Test framework | Flexible and Pythonic |
| Postman | API test authoring | Visual and scriptable API tests |
| Newman | API test runner | Enables CLI and CI integration |
| dotenv | Config management | Securely handles credentials |

5. How to Run the Tests

UI Tests

Set environment variables in `.env`:

```
URL=http://localhost:5173/login?redirect=%2Ftodo
# Valid and Invalid credentials for testing login functionality
VALID_EMAIL=john@example.com
VALID_PASSWORD=Password123
INVALID_EMAIL=johen@example.com
INVALID_PASSWORD=Password123
```

Run tests:

```
cd tests/selenium_tests
pytest
```

API Tests

Open Postman and verify: - `todo-tests.postman_collection.json` -
`todo-env.postman_environment.json`

Run via CLI:

```
newman run todo-tests.postman_collection.json -e todo-env.postman_environment.json
```

6. Assumptions & Limitations

- API routes for PUT and DELETE are not implemented; corresponding tests are skipped or commented out.
- No database resets between test runs; tests may require cleanup manually or may fail on re-run due to duplicate entries.

7. Project Structure Overview

```
tests/
  api_tests/
    todo-env.postman_environment.json
    todo-tests.postman_collection.json
  selenium_tests/
    tests/
      crud_test.py
      login_test.py
    utils/
      base_test.py
  requirements.txt
```