**CS673 Software Engineering** 

**Team 3 - MyMedic**

**Software Test Document**

| Team Member | Role(s) | Signature | Date |
| --- | --- | --- | --- |
| Adriel Domingo | Requirements | *Adriel Domingo* | 5/26/2025 |
| [John Gutierrez](mailto:johng13@bu.edu) | All | [*John Gutierrez*](mailto:johng13@bu.edu) | 6/9/2025 |
| Mengliang Tan | QA | *Mengliang Tan* | 5/26/2025 |
| Uzay Isin Alici | Security | *Uzay Isin Alici* | 5/26/2025 |
| Tyler Gonsalves | Configuration/ Design | *Tyler Gonsalves* | 5/26/2025 |
| Indra Sigicharla | Team/Configuration | *Indra Sigicharla* | 5/26/2025 |
| Hongcheng Ding | Design | *Hongcheng Ding* | 5/26/2025 |
|  |  |  |  |

**Revision history**

| **Version** | **Author** | **Date** | **Change** |
| --- | --- | --- | --- |
| **1.0** | **Team 3** | **5/26/2025** | **N/A** |
|  |  |  |  |

[Testing Summary](#_sm5odwyvuk3j)

[Manuel Tests Reports](#_pqso2mbjyzx4)

[Automated Testing Reports](#_mtfbusfb0eq3)

[Testing Metrics](#_rijyjeu2ojqa)

[References](#_15tmymhipvdv)

[Glossary](#_8n34lvocupub)

# Testing Summary

## Unit Testing

Unit testing will be performed upon every check-in and merge to the main branch to ensure that the master branch is always in a runnable state. Unit testing will also be run on push to any branch. The results of these tests will be recorded in the artefacts of every release for tracking. Unit tests will be incrementally added as new features are developed.

## Integration Testing

Integration testing will be performed to verify API functionality between the frontend and backend, and between backend components. These tests will be performed upon merges to the main branch to validate system functionality on the main branch.

## System Testing

System testing will be performed manually prior to each release of the code. The test results will be included in the release notes for tracking.

## Acceptance Testing

Acceptance testing will be performed manually on each merge of the code to the main branch to validate added features. The test results will be included in the pull request for merging the feature branch to document feature functionality.

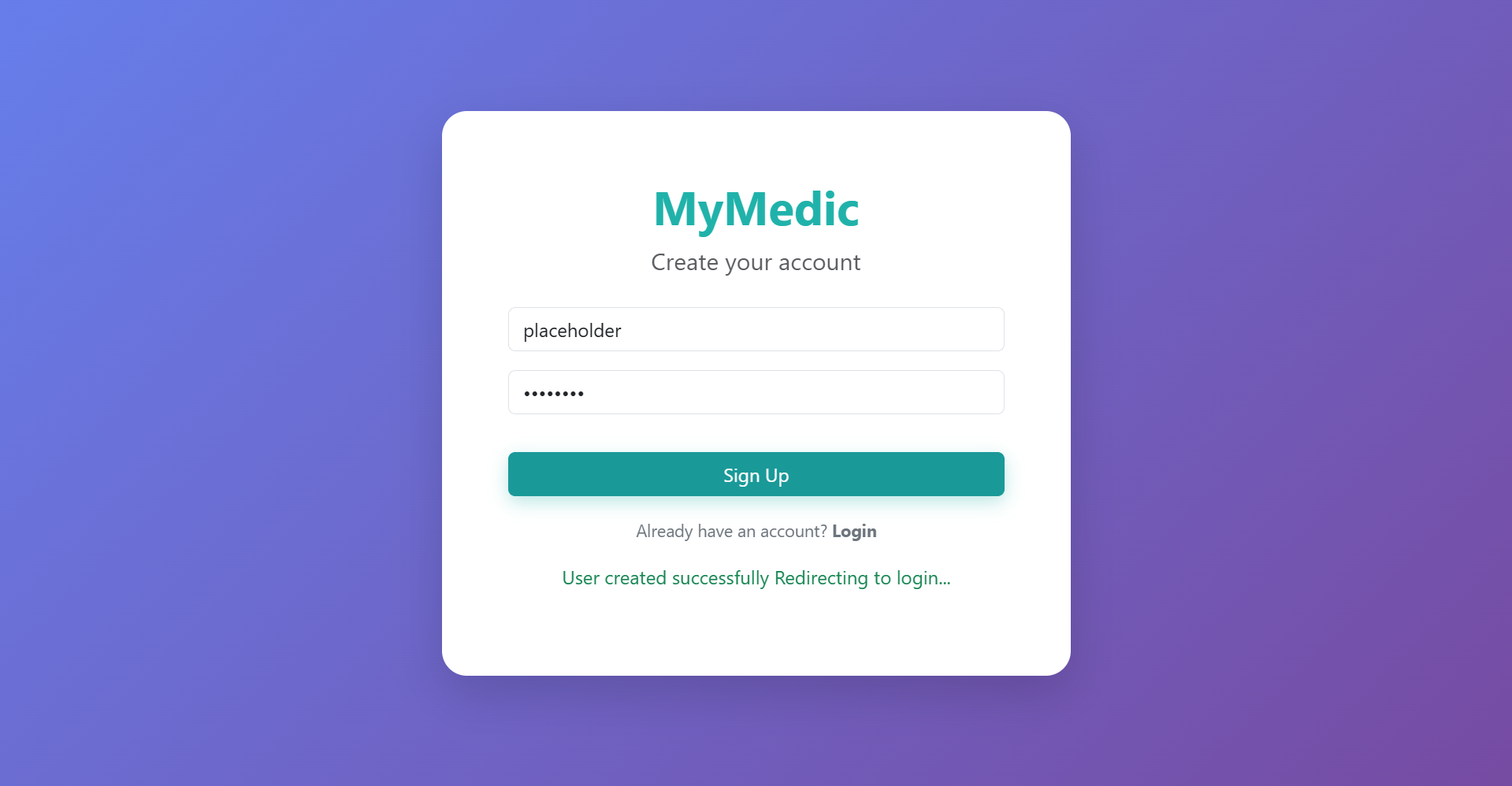
## Regression Testing

Regression testing will be performed on every push to every branch. As defects are added and fixed, tests to verify the bug is resolved will be added to the regression test suite. Regression test results will be recorded in pull requests and in releases of code.

# Manual Testing Report

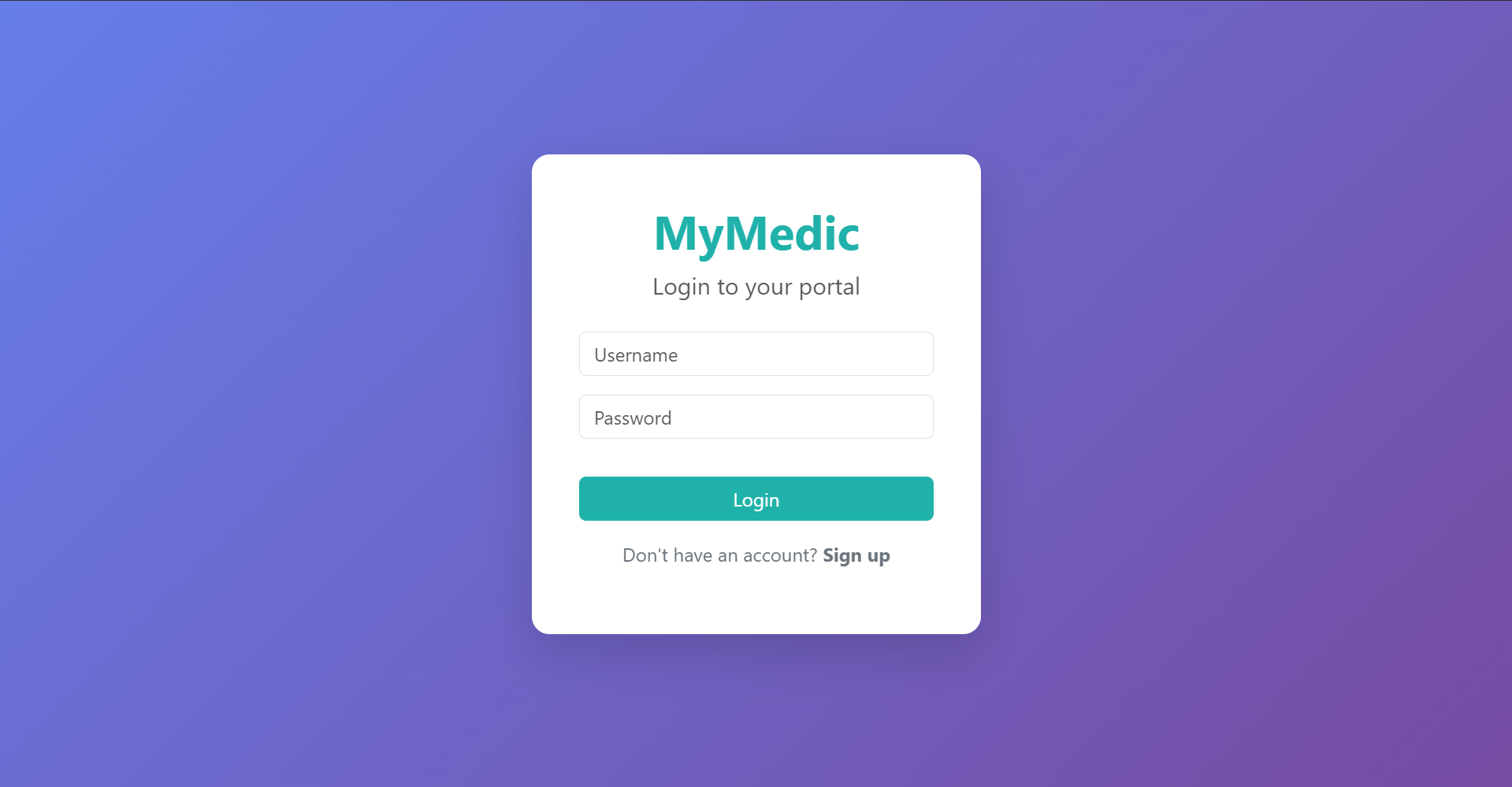
## Test Case 1: User Registration

* New or old: New
* Test items: User Registration Page
* Test priority High
* Dependencies N/A
* Preconditions: Server is running
* input data:
  + Username: testuser
  + Password: 6uj9rD46!L#b
* Test steps:
  + Enter username in the username field of the registration form
  + Enter the password in the password field of the registration form
  + Click the “Sign Up” button
* Postconditions: Success message printed at the bottom of the form
* Expected output: User redirected to the login screen
* Actual output: User redirected to the login screen
* Pass or Fail: pass
* Bug id/link: <https://github.com/BUMETCS673/CS673OLSum25Team3/issues/11>
* Additional notes:



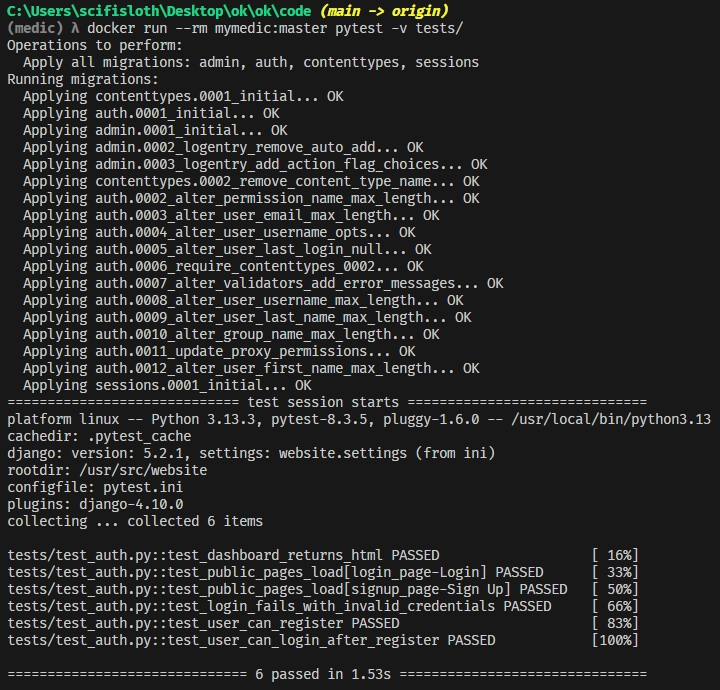
## Test Case 2: User Login

* New or old: New
* Test items: User Registration Page
* Test priority High
* Dependencies Success of Test Case 1
* Preconditions:
  + Server is running
  + User is registered
* input data:
  + Username: testuser
  + Password: 6uj9rD46!L#b
* Test steps:
  + Enter username in the username field of the registration form
  + Enter the password in the password field of the registration form
  + Click the “Login” button
* Postconditions: N/A
* Expected output: User redirected to the home page
* Actual output: User redirected to the home page
* Pass or Fail: pass
* Bug id/link: <https://github.com/BUMETCS673/CS673OLSum25Team3/issues/11>
* Additional notes:



# Automated Testing Report

Automated unit and regression testing will be completed using pytest and jest for testing the python and javascript code, respectively. Cypress and Selenium will be used to complete integration, and system testing where appropriate. These tests will be orchestrated using github actions.



# Testing Metrics

In this section, you shall report any metrics used for the evaluation, e.g. # of test cases, test coverage, defects rate, etc.

| **Metric Name** | **Description** |
| --- | --- |
| Test Cases | Number of unique tests executed |
| Test Coverage | Initial statement coverage will be leveraged to ensure dead code is kept to a minimum. As the development progresses the coverage will be shifted to path coverage to better capture the flow of code during runtime. |
| Test case pass rate | Pass rate must be greater 80% |
| Defect rate | 20 bugs or less per 1000 lines of code (KLOC) |

| **Metric Name** | **Description** |
| --- | --- |
| **Test Cases** | 6 unique test cases executed as shown in the output from pytest -v tests/. |
| **Test Coverage** | Currently executing statement-level tests like user registration, login, dashboard access, and page loads. As development progresses, this can be expanded to path coverage by including tests that validate different conditional flows and edge cases in views, forms, or APIs. |
| **Test Case Pass Rate** | 100% pass rate (6/6 passed), which is well above the 80% threshold. |
| **Defect Rate** | No test failures or defects detected in the current suite. Assuming fewer than 20 known bugs per 1000 lines of code (KLOC), this passes the given quality threshold. |

# References

# Glossary