**CS673 Software Engineering**

**Team 1 - Menu Match**

**Software Test Document**

| Team Member | Role(s) | Signature | Date |
| --- | --- | --- | --- |
| Roshni Dodhi | Requirement Leader |  | 12/08/2024 |
| Prayushi Khandelwal | Security Leader | *P.K* | 11/07/2024 |
| Jiho Cheon | QA Leader | *Jiho Cheon* | 12/08/2024 |
| Vignesh Swaminathan | Configuration Leader | *vignesh* | 12/08/2024 |
| Anshul Raj | Design and Implementation Leader | AR | 11/07/2024 |
| Dirgha Jivani | Team Leader | *Dirgha* | 12/08/2024 |
| Pratyush Patel | Design and Implementation Leader | *Pratyush* | 12/08/2024 |
|  |  |  |  |

**Revision history**

| **Version** | **Author** | **Date** | **Change** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

[Testing Summary](#_heading=h.gjdgxs)

[Manuel Tests Reports](#_heading=h.30j0zll)

[Automated Testing Reports](#_heading=h.1fob9te)

[Testing Metrics](#_heading=h.3znysh7)

[References](#_heading=h.2et92p0)

[Glossary](#_heading=h.tyjcwt)

# Testing Summary

The testing phase covered several aspects of the system, focusing on functionality, integration, and robustness. The following testing techniques were utilized:

* **Unit Testing**: Conducted for core functionalities such as user registration, login, and route protection to ensure the components work as expected in isolation.
* **Integration Testing**: Verified the correct interaction between various modules, ensuring that endpoints and database interactions were seamless.
* **System Testing**: Evaluated the entire system to ensure it meets the specified requirements and functions correctly as a whole.
* **Acceptance Testing**: Ensured that the developed functionalities met the requirements and expectations outlined by the project stakeholders.
* **Regression Testing**: Verified that recent code changes did not negatively impact the existing functionalities.

**Testing Tools**:

* **Postman** was used for API testing to validate endpoints, their responses, and error handling.

# Manual Testing Report

In this section, you will give a detailed description of each manual test case performed and the result. If this is a previous You shall list what are existing tests developed in the previous semester and what are new tests developed currently.

| **Test Case ID** | **API Endpoint** | **Description** | **Method** | **Input Data** | **Expected Output** | **Actual Output** | **Status** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TC-01 | /api/auth/signup | Test user signup with valid data | POST | { "username": "testuser", "email": "test@example.com", "password": "securepassword" } | {"message": "User registered successfully"} |  | Pass |  |
| TC-02 | /api/auth/signup | Test signup with existing email | POST | { "username": "newuser", "email": "test@example.com", "password": "newpassword" } | {"error": "Email already registered"} |  | Pass |  |
| TC-03 | /api/auth/login | Test login with valid credentials | POST | { "email": "test@example.com", "password": "securepassword" } | { "access\_token": "JWT\_ACCESS\_TOKEN", "user\_id": 1 } |  | Pass |  |
| TC-04 | /api/auth/login | Test login with invalid credentials | POST | { "email": "test@example.com", "password": "wrongpassword" } | {"error": "Invalid credentials"} |  | Pass |  |
| TC-05 | /api/auth/logout | Test logout with valid JWT | POST | Header: Authorization: Bearer JWT\_ACCESS\_TOKEN | {"message": "Logged out successfully"} |  | Pass |  |
| TC-06 | /api/users/ | Retrieve all users (admin access) | GET | - | List of user objects |  | Pass |  |
| TC-07 | /api/users/<user\_id> | Retrieve a specific user by ID | GET | <user\_id> = 1 | User object with user\_id 1 |  | Pass |  |
| TC-08 | /api/restaurants/ | Get all restaurants | GET | - | List of restaurant objects |  | Pass |  |
| TC-09 | /api/restaurants/<restaurant\_id> | Get a restaurant by ID | GET | <restaurant\_id> = 1 | Restaurant object with restaurant\_id 1 |  | Pass |  |
| TC-10 | /api/reviews/ | Get all reviews | GET | - | List of review objects |  | Pass |  |
| TC-11 | /api/reviews/<review\_id> | Get a review by ID | GET | <review\_id> = 1 | Review object with review\_id 1 |  | Pass |  |

### **Summary**

* + **Total Test Cases**: 11
  + **Pass Rate**: 100%
  + **Test Coverage**: Comprehensive coverage of CRUD operations and authentication flows

# Automated Testing Report

Describe briefly the automated testing you have done, including where the test code resides in your code repository, what test frameworks are used, and the screen shots or generated testing report.

# Testing Metrics

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

# References

# Glossary

* + **API Testing**: Verifies that application programming interfaces (APIs) function correctly.
  + **Unit Testing**: Focuses on individual components in isolation.
  + **Integration Testing**: Ensures different modules work together properly.
  + **System Testing**: Verifies the system as a whole meets requirements.
  + **Acceptance Testing**: Confirms the system meets business requirements.
  + **Regression Testing**: Checks that recent code changes do not break existing functionality.