Test Plan Document:

1. Introduction:

- Purpose: The purpose of this test plan is to outline the testing approach, scope, and resources required for testing the graphical user interface of the UniCourt support portal.

- Objectives: The objectives of the testing are to ensure the functionality, usability, and performance of the support portal.

2. Scope:

- In-Scope:

- Authentication and authorization functionality.

- User roles and permissions.

- Access control to different pages based on user roles.

- Uptime and scalability requirements.

- Out-of-Scope:

- Back-end database testing.

- Testing of external integrations (Google OAuth2, Social Login).

3. Test Approach:

- Testing Levels: The testing will primarily focus on system testing.

- Testing Types: Functional testing, Usability testing,Compatibility Testing, Performance testing.

- Test Techniques: Black-box testing, Boundary value analysis, Exploratory Testing, Positive/negative testing.

4. Test Environment:

- Operating System: Windows 10.

- Browsers: Chrome, Firefox, Safari.

- Test Case Management Tool: TestLink.

- Defect Tracking Tool: Jira.

5. Test Schedule:

- Requirement Analysis and Test Planning: Week 1.

- Test Case Preparation and review: Week 2.

- Test Execution: Week 3.

- Test Reporting: Week 4.

6. Test Cases:

- Test cases will be prepared to validate each functional requirement mentioned in the SRS document.

- Test cases will cover positive and negative scenarios, boundary values, and different user roles.

7. Test Execution:

- Test cases will be executed according to the planned schedule.

- Defects will be logged in Jira with appropriate priority and severity levels.

- Test data will be prepared to simulate different user roles and access scenarios.

8. Test Reporting:

- Test results will be documented in TestLink.

- Test summary reports will be prepared to provide an overview of the testing activities and outcomes.

9. Risks and Mitigation:

- Risks:

- Insufficient testing resources.

- Delays in development impacting testing schedule.

- Mitigation:

- Regular communication with the development team to align schedules.

- Prioritization of test cases based on critical functionality.