Test Plan (ELTS Application)

1. Objective

The objective of this Test Plan is to define the strategy for testing the ELTS (English Language Testing System) application. The goal is to ensure the application functions as expected across all features, focusing on user registration, login, dashboard interaction, practice modules, and integration with Google for authentication.

- To validate that all features and functionalities work as per the requirements.
- To ensure user data is handled correctly, including registration and login.
- To test the practice modules and their performance.
- To verify compatibility across multiple devices and browsers.

Technology Stack:

- React (Version 18.2.0)
- jQuery 2.1.1
- JavaScript
- PostgreSQL (Database)
- Web Servers (Apache, Nginx)

---

2. Scope

The scope includes testing the following features and functionalities of the ELTS application:

- **In-Scope**:
    - User Registration (including validation of phone numbers and qualification details)
    - Login functionality (Google login and manual login)
    - Dashboard Page (viewing past practices and updating profile)
    - Practice Module (starting and completing practice tests)
    - Mobile and desktop compatibility for user interface and performance
- **Out-of-Scope**:
    - Any third-party integrations beyond Google login
    - Backend services unrelated to the core application functionality
    - Non-core user-facing features (e.g., support page or chat)

---

3. Inclusions

This section covers the key features to be tested:

- **Features**:
    - User registration and login
    - Phone number verification
    - Qualification page and validation
    - Profile management (updating profile)
    - Practice modules and tracking
    - Mobile responsiveness and UI consistency

---

4. Exclusions

The following features are out of scope for testing:

- Third-party integrations (except Google Login)
- Support page or live chat widget

---

5. Test Environments

Testing will be conducted in the following environments:

Environment	URL	Platform
		Windows 10, macOS, Mobile OS
		Windows 10, macOS, Mobile OS
		Windows 10, macOS, Mobile OS
		Windows 10, macOS, Mobile OS

Supported Environments:

- **Operating Systems**: Windows 10, macOS, Android, iPhone
- **Browsers**: Chrome, Firefox, Edge, Safari

---

6. Defect Reporting Procedure

- **Defect Identification**: Defects will be identified based on deviations from expected functionality or user experience issues.

- **Reporting Process**: Defects will be reported through JIRA with detailed steps for reproduction, logs, and screenshots.
- **Defect Severity**: Defects will be categorized (Critical, High, Medium, Low) and assigned for resolution.
- **Communication**: Daily status updates will be communicated to the development management team regarding defect resolutions.

POC for Defects:

- Frontend: [Assign Name]
- Backend: [Assign Name]
- DevOps: [Assign Name]

---

7. Test Strategy

The testing approach will involve:

- **Test Design Techniques**:
    - Equivalence Class Partitioning
    - Boundary Value Analysis
    - Decision Table Testing
    - State Transition Testing
    - Use Case Testing
- **Testing Types**:
    - **Smoke Testing**: To verify core functionalities such as login, registration, and dashboard access.
    - **Sanity Testing**: To verify that new code changes work as expected without breaking existing features.
    - **Regression Testing**: To ensure that new functionalities do not disrupt existing features.
    - **Usability and UI Testing**: To check the user interface for usability and accessibility.
    - **Functional Testing**: To test the individual features like registration, login, profile updates, etc.
    - **Cross-Browser Testing**: To verify the application works across different browsers.
    - **Mobile Testing**: To ensure mobile responsiveness and functionality.

---

8. Test Schedule

- **Creating Test Plan**: [Date]

- **Test Case Creation**: [Date]
- **Test Case Execution**: [Date]
- **Summary Reports Submission**: [Date]

Testing will be executed in two sprints, ensuring that all key features are covered.

---

9. Test Deliverables

- Test Plan Document
- Test Cases and Test Scenarios
- Test Execution Reports
- Defect Reports
- Final Test Summary Report

---

10. Entry and Exit Criteria

Entry Criteria:

- Requirements and specifications are received and understood.
- Test environment is set up and accessible.
- Test cases and test data are prepared.

Exit Criteria:

- All test cases are executed.
- No critical defects remain unresolved.
- Final Test Summary Report is prepared and shared with stakeholders.

---

11. Test Execution

Entry Criteria:

- Test cases and scenarios are signed off by the client.
- Application is stable and ready for testing.

Exit Criteria:

- Test case reports and defect reports are finalized and delivered.

---

12. Test Closure

Entry Criteria:

- Test execution is complete.
- All defects are resolved or documented.

Exit Criteria:

- Test Summary Report is prepared and shared with stakeholders.

---

13. Tools

The following tools will be used for the project:

- **ClickUp** for defect tracking
- **Mind Map Tool** for test planning
- **Snipping Tool** for screenshots
- **Excel/Word** for documentation

---

14. Risks and Mitigations

Risk	Mitigation
Non-availability of resources	Backup resource planning
Build URL not working	Work on other tasks while waiting for the stable build
Insufficient time for testing	Ramp up resources dynamically based on client needs

---

15. Approvals

Testing will proceed once the following documents are approved by the client:

- Test Plan
- Test Scenarios
- Test Cases
- Reports