Test Report for FloTorch.ai

---

1. Overview

This report provides a comprehensive evaluation of FloTorch.ai's functionality, performance, and reliability as a Retrieval-Augmented Generation (RAG) evaluation toolkit. The analysis includes testing critical features, configurations, and workflows while validating system stability and compliance with expected outcomes.

---

2. Project Information

- **Project Name**: FloTorch
- **Version**: 1

---

3. Objectives

The primary objectives of this testing phase are:

- **Functional Validation**: Verify the operational correctness of all features.
- **Performance Testing**: Measure the robustness and efficiency of core processes, including experiment execution and evaluation metrics.
- **Feature Testing**: Validate critical user flows such as project creation, configuration uploads, and experiment tracking.
- **Metrics Accuracy**: Ensure alignment of evaluation metrics (faithfulness, context precision, aspect critic, and answer relevancy) with expected outputs.

---

4. Testing Approach

Testing was performed across the following scenarios:

1. **Installation and Deployment**:
    - Verified the installation process on AWS EC2 instances using provided scripts (`deploy.sh`).
    - Confirmed prerequisites and infrastructure components, including VPC, DynamoDB, ECS, and App Runner.

2. **Functional Testing**:
    - Created projects with multiple configurations.
    - Tested indexing and retrieval strategies (e.g., chunking strategy, embedding models, KNN settings).
3. **UI Validation**:
    - Navigated through the FloTorch UI to create and manage projects.
    - Assessed the workflow for saving/loading configurations.
4. **Evaluation Accuracy**:
    - Verified automated evaluation metrics against expected results for multiple LLM configurations.

---

5. Test Summary

- **Total Test Cases Executed**: 69
- **Passed:** 68
- **Failed: 01**
- **Test Environment**:
    - OS: Windows 11
    - Browser: Google Chrome

---

6. Defect Categorization

- **Defects Identified**: 48
    - **High Priority**: 10
    - **Medium Priority**: 10
    - **Low Priority**: 28

7. Known Issues

1. **Missing Exception Handling**:
    - If a model is unavailable in the Bedrock configuration, the system lacks exception handling, resulting in errors when calculating costs.
2. **Project Status Management**:
    - If a Step Function aborts or times out mid-project, the system fails to update the status, requiring manual intervention.
3. **File Upload Completeness**:
    - Issues encountered when executing hierarchical chunking strategies through configuration uploads.

8. Key Findings

1. **Strengths**:
    - **Automated Setup**: Deployment scripts simplify infrastructure provisioning and application setup.
    - **UI Functionality**: Intuitive interface with clear navigation paths for creating, managing, and monitoring experiments.
    - **Scalability**: Efficient handling of multiple concurrent experiments with robust state management.
2. **Areas for Improvement**:
    - **Error Handling**:
        - Missing exception handling when unsupported configurations are uploaded.
        - Timeout issues in AWS Step Functions occasionally leave projects in an inconsistent state.
    - **Documentation Gaps**:
        - The FAQs lack clarity on some key aspects, such as troubleshooting experiment failures.
    - **Performance Bottlenecks**:
        - Longer-than-expected execution times for larger datasets.

---

9. Recommendations

1. **Exception Handling**:
    - Introduce exception blocks to gracefully handle unsupported configurations, ensuring uninterrupted execution.
2. **Project Recovery Automation**:
    - Implement mechanisms to auto-recover from Step Function failures or provide detailed logs for quick troubleshooting.
3. **Enhanced Documentation**:
    - Expand FAQs to address troubleshooting and advanced usage scenarios.
4. **Performance Optimization**:
    - Investigate execution time delays and optimize retrieval strategies for larger datasets.

---

10. Conclusion

FloTorch.ai demonstrates strong potential as a reliable RAG evaluation toolkit, with robust automation, intuitive workflows, and effective performance metrics. Addressing the identified

issues will ensure a more seamless user experience and enhance system reliability in production environments.