Date: December 2,2025

RAPTOR KIDZ – ULTRA-DETAILED PROJECT MONTHLY CLOSURE REPORT – NOVEMBER 2025

EXECUTIVE SUMMARY

Raptor KIDZ entered the month with major dependencies on abandoned legacy technologies and undefined cloud infrastructure, yet the team delivered consistent progress across all technical layers—reverse engineering, architecture definition, feasibility validation, cloud cost analysis, UI planning, backend structuring, and controlled development cycles.

Despite tight constraints, uncertain legacy formats, and platform discontinuations (Flash), the team achieved successful rendering through Ruffle, providing a functional and testable APK. This has positioned the project for a strong December execution sprint involving cloud provisioning, UI continuity, backend completion, and integration.

PHASE I – LEGACY SYSTEM ANALYSIS & REPOSITORY REALIGNMENT

Objective: Understand existing assets, identify constraints, determine salvageable components.

1. **Initial Environment Review:**
   - Project inherited with a partially documented structure built on **.NET (C#)** and **SQL Server**, requiring reconstruction of module-level dependencies.
   - Code pathways had minimal mapping to front-end deliverables, increasing the need for deep analysis.
2. **Repository Audit:**
   - Primary repository contained **.swf** files—Flash-based, highly outdated, and dependent on Adobe Flash Player.
   - No direct source code versions (AS2/AS3) available, increasing reverse-migration complexity.

3. **Critical Technical Blockage Identified:**
   - As **Flash was globally discontinued**, the team could not preview or extract SWF functionalities through conventional emulators.
   - This forced a search for alternative rendering solutions.
4. **Alternate Rendering Solution Discovery:**
   - After evaluating multiple fallback renderers, **Ruffle**, an open-source Flash Player emulator written in Rust, emerged as the only feasible candidate.
   - Initial scope included compatibility mapping, performance benchmarking, and isolation of unsupported SWF operations.

PHASE II – RUFFLE FEASIBILITY STUDY & FUNCTIONAL DECOMPOSITION

Objective: Validate whether Ruffle can reliably replace Flash without loss of functionality.

1. **Ruffle Technical Study by Mr. Yashwanth & Team:**
   - Deep-dive into WebAssembly behavior, AVM1/AVM2 support, asset loading, and interaction layers.
   - Identified constraints including limited ActionScript 3 compatibility and animation timing inconsistencies.
2. **Functional Breakdown:**
   - Each SWF file was decomposed functionally—animations, button events, audio triggers, timeline sequences, and asset loaders.
   - Compatibility index prepared for each module.
3. **Complexity Identification:**
   - Complex areas included:
     - Multi-layered interactive timelines
     - Legacy ActionScript calls
     - Performance drops in WebView containers
     - Synchronization between internal object states
   - These issues required senior technical review.

4. **Technical Guidance from Mr. Nataraj:**
   - Conducted a structured feasibility session.
   - Provided parameter-based evaluation:
     - Render stability
     - Memory consumption
     - Frame skip tolerance
     - UI overlay feasibility
     - Ruffle patch-level fallback
   - Helped team prioritize achievable modules and isolate high-risk ones.

PHASE III – RUFFLE IMPLEMENTATION, WEBVIEW VALIDATION & TEST APK BUILD

Objective: Confirm Ruffle integration on mobile environment and deliver a working proof.

1. **Ruffle Integration Success:**
   - Ruffle successfully embedded into mobile WebView containers.
   - Frame rendering stabilized after iterative adjustments.
2. **WebView Container Optimization:**
   - Evaluated container-level behavior for:
     - Touch responsiveness
     - Memory handling on mid-range devices
     - Offline caching
     - File access permissions
   - Achieved stable performance after adjusting resource loading logic.
3. **Test APK Submission:**
   - A fully compiled **sample Test APK** created.
   - Validated by internal reviewers and confirmed as:
     - Functionally viable
     - Render-complete
     - Free from major runtime exceptions

PHASE IV – UI WALKTHROUGH, ARCHITECTURAL DEBATE & STRUCTURE FINALIZATION

Objective: Align on UI structure, validate content categorization model, finalize architecture.

1. **UI Review Session with Mr. Raja:**
    - Invited to review UI flow, content layering, and screen hierarchy.
2. **Team Walkthrough Execution:**
    - Demonstrated user flow: Splash → Login → Dashboard → Content Categories → SWF Rendering Screen.
3. **Architecture Change Proposal by Mr. Raja:**
    - Recommended compressing entire content categorization into **one universal screen** for simplified navigation.
    - This conflicted with content depth, multi-level subjects, and future scalability.
4. **Technical Evaluation of Proposed Change:**
    - Team identified the following conflicts:
        - Loss of hierarchical clarity
        - High UI congestion
        - Increased load time per screen
        - Unmanageable asset preloading
    - Concluded that single-screen condensation was **technically non-viable** for Ruffle-based rendering.
5. **Decision:**
    - Architecture change dropped after discussion.
    - Reinforced continuation with the **multi-layer structured architecture**, ensuring clarity and future-proofing.

PHASE V – BASE ARCHITECTURE BUILD, FEATURE INITIATION & CLOUD STUDY

Objective: Start app-level feature development and determine long-term cloud strategy.

1. **Architecture Build-out:**
    - Core shell created for the app.
    - Navigation stack finalized.

2. **Feature Initiation:**
    - Login system drafted
    - Profile settings structured
    - Ruffle container integration planned inside main application flow
3. **Cloud Research Framework:**
    - Compared four major platforms on:
        - Cost
        - Elastic scalability
        - Security
        - CDN availability
        - Long-term affordability
    - Platforms: Azure, AWS, GCP, DigitalOcean
4. **Cloud Recommendation:**
    - **Google Cloud Platform** chosen due to:
        - Lower long-term storage & transfer costs
        - Superior multimedia optimization
        - Integration compatibility with Raptor AI's existing infra
        - Projected future fuel cost impact

PHASE VI – CLOUD SECURITY MODEL, COST ANALYSIS & INTERNAL ALIGNMENT

Objective: Create a secure and cost-efficient cloud environment.

1. **Requirement Analysis:**
    - Jointly executed by **Mr. Selva** and **Mr. Nataraj**, referencing Raptor AI's security model.
2. **Cloud Allocation Blueprint:**
    - Designed with strong emphasis on:
        - Least privilege IAM policies
        - Asset segregation in bucket layers
        - Network firewall restrictions
        - Cost-protected resource allocation
3. **Delay Due to Audit & Cross-Project Analysis:**

- Internal cloud audit triggered cross-verification due to Raptor AI's parallel infra expansion.
- Caused necessary but unavoidable delay to prevent resource conflicts, cost wastage, and redundant provisioning.

PHASE V (EXTENDED) – DELAY COMMUNICATION & LOCAL DEVELOPMENT BACKUP

Objective: Ensure continuity and prevent impact from cloud delays.

1. **Delay Communication:**
   - Immediately informed to:
     - Mr. Yashwanth
     - Full development team
     - Raptor Technologies management
2. **Fallback Plan Execution:**
   - Under guidance from **Mr. Nataraj**, team resumed active development locally using mock cloud structures.
   - Prevented project slowdown and ensured continuous progress.

PHASE VI (EXTENDED) – FINAL OUTPUT SUBMISSION & DECEMBER PREP

1. **Raja Discussion Scheduled:**
   - First week of December.
2. **Bucket File Distribution Plan:**
   - Team to receive support from **Mr. Gurumoorthy** and **Mr. Nataraj** for correct bucket allocations.
3. **Current Output Validation:**
   - Submitted to **Mr. Nataraj**, who confirmed the output quality and approved progression.

DECEMBER EXECUTION PLAN:

1. UI Full Implementation
2. Backend Completion
3. Cloud Allocation by Mr. Selva & Mr. Nataraj

APPRECIATION & POSITIVE FEEDBACK

Raptor KIDZ Student Project Team Appreciation – Ruffle Achievement

- The entire team deserves strong recognition for achieving a working Ruffle-based solution despite dealing with Open source complexities.
- Delivering a **functional Test APK under such constraints** demonstrates strong problem-solving ability, resilience, and technical grit.

Appreciation for Mr. Nataraj – Technical Direction & Stability

His leadership was instrumental in:

- Breaking down SWF complexities
- Providing clear feasibility paths
- Guiding fallback planning
- Mentoring the team through architectural and rendering challenges
- Ensuring development continuity during cloud delays

His involvement directly ensured the project's momentum didn't break at any stage.

Appreciation for Mr. Selva & Mr. Nataraj – Cloud Infrastructure Excellence

Their cloud evaluation was thorough, financially responsible, and strategically aligned. Key strengths in their approach:

- High cost savings
- Risk-averse planning
- Scalable architecture
- Future-proof design

- Strong alignment with Raptor AI's long-term infra

They ensured the company avoided unnecessary cloud expenditure while maintaining top-tier security and performance standards.

Acknowledgment for Mr. T. S. Karthikeyan – Budget Approval

His timely approval of the optimized cloud budget ensured the project continued without operational pauses. This supported the long-term economic safety of the project.

Thank You Note – CIT Management, Mrs. Shanmughapriya & Student Team

A heartfelt thanks to **Mrs. Shanmughapriya**, the student team, and the **CIT Management** for supporting research activities, providing technical bandwidth, and enabling smooth institutional collaboration. Their contribution helped accelerate early-stage groundwork and strengthened academic partnership value.

Warm Regards,

Raptor Technologies