SafeData 2.0 - Project Handover Package

This document serves as a complete handover package for the SafeData 2.0 project. It contains the research, logic, architecture, methodology, and implementation strategy for the project, along with guidelines for prototype development. SafeData 2.0 is an advanced federated learning-based data privacy solution that ensures user data remains secure and anonymized during processing.

1. Research & Idea

SafeData 2.0 leverages federated learning to train models without centralizing raw data. Instead, model training occurs locally on user devices, and only anonymized model parameters are shared with the central server. This approach significantly reduces privacy risks while maintaining machine learning performance. Key Research Areas: - Federated Learning frameworks (TensorFlow Federated, PySyft, Flower) - Data anonymization techniques (Differential Privacy, k-Anonymity) - Encryption methods for model parameter transmission - Secure aggregation protocols

2. Methodology

Step 1: Data stays on the device (client-side processing) Step 2: Local model training is performed on anonymized data Step 3: Encrypted/anonymized model parameters are sent to the server Step 4: Secure aggregation of updates to produce a global model Step 5: Dashboard visualizes aggregated insights without exposing personal data

3. System Architecture

The SafeData 2.0 architecture consists of: - Client-Side Application: For local data processing & model training - Secure Communication Layer: For encrypted parameter exchange - Central Aggregation Server: For secure federated aggregation - Dashboard/Frontend: For visualization of anonymized results

4. Implementation Plan

Phase 1: Research & Requirement Gathering Phase 2: Backend Development (Federated Learning & Aggregation) Phase 3: Frontend Development (React/Next.js + Visualization) Phase 4: Integration & Testing Phase 5: Prototype Launch

5. Prototype Development

The prototype will simulate a federated learning environment with multiple clients. A mock dashboard will be developed to showcase real-time visualizations of anonymized aggregated data. Frontend will be built using Next.js and React, while backend will use Python and FL frameworks.

6. Output Format

The system outputs aggregated analytics in visual form via charts, graphs, and tables in the dashboard. Raw or sensitive user data will never leave the client device.