Software Test Plan (STP)

# Project: 3D Fashion Data Pipeline

## 1. Introduction

This Software Test Plan outlines the testing strategy for the 3D Fashion Data Management System. The goal is to ensure correct conversion of clothing items into GLB format, successful storage in MongoDB and PostgreSQL, metadata integrity, and rendering compatibility with web-based and external 3D viewers.

## 2. Test Items

- Trimesh-based Conversion Module (`trimesh\_converter.py`)  
- Dataset Processor (`DatasetProcessor`)  
- PostgreSQL Uploader (`PostgresUploader`)  
- MongoDB Uploader (`MongoUploader`)  
- Flask API Server (`api\_server.py`)  
- GLB Viewer Integration (via https://gltf-viewer.donmccurdy.com)

## 3. Features to be Tested

- Conversion of .obj files (with textures) into .glb  
- Recursive processing of dataset directories (including nested folders)  
- Correct metadata extraction and upload to PostgreSQL  
- Binary storage of GLB in MongoDB GridFS with item ID binding  
- Retrieval of GLB via Flask API (`/api/glb/<item\_id>`)  
- Viewing compatibility in external GLTF viewer

## 4. Features Not to be Tested

- Advanced styling or customization in the GLB viewer  
- AI-based recommendation of clothing items (not implemented)  
- Authentication or user-level personalization

## 5. Testing Strategy

- Unit Testing: Test converter and uploader classes in isolation using controlled OBJ inputs  
- Integration Testing: Simulate full flow from directory input → GLB generation → database upload  
- System Testing: Full run using `main.py` with multiple clothing categories  
- Acceptance Testing: Manual verification using external GLTF viewer and database queries

## 6. Test Environment

- OS: Windows 10  
- Backend: Python 3.11  
- Libraries: `trimesh`, `flask`, `psycopg2`, `pymongo`, `gridfs`  
- Database: PostgreSQL 15, MongoDB 6  
- Browser: Chrome (latest) for GLB preview  
- External Tool: https://gltf-viewer.donmccurdy.com

## 7. Responsibilities

## - Golan: System architecture, trimesh conversion logic, GLB validation, and PostgreSQL upload testing.

## - Dana: Dataset processing, MongoDB upload validation, Flask API testing, and external viewer verification.

## 8. Schedule

- Unit & Conversion Testing: Week 1  
- Integration Testing: Week 2  
- Full Dataset Testing & DB Validation: Week 3  
- Acceptance (Manual) + Viewer Verification: Week 4

## 9. Risks and Contingencies

- ❗ Missing textures in OBJ/MTL may cause blank GLB renders  
 → fallback to default material and log warning  
- ❗ Directory paths may include corrupted or incomplete files  
 → add try-except wrappers with error logging  
- ❗ PostgreSQL schema mismatch  
 → rollback to reset schema before new inserts